The buckling of grid shells

Mohamedien, Mohamed Ahmed

Award date:
1990

Awarding institution:
University of Bath

THE BUCKLING OF GRID SHELLS

Submitted by **Mohamed Ahmed Mohamedien**
B.Sc.(Eng), M.Sc.(Eng.)
for the degree of
Doctor of Philosophy
of the University of Bath
1990

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

To the memory of my late father,
my mother and my wife

ABSTRACT

A new finite element technique based on the bicubic B-spline element is introduced for the prediction of the buckling behaviour of grid shells. Physical model test results show a good agreement with the numerical analysis.

ACKNOWLEDGEMENTS

The author wishes to express his deepest thanks to his supervisor, Mr. C.J.K. Williams, School of Arch. and Building Eng., University of Bath, for his help, cooperation, and valuable guidance, as well as his constant encouragement throughout this work. Words, however ample, would be insufficient to express the author's gratitude to him.

The author would like to thank Dr. A.J.Wilson and Mr. S.G.Ring for their support and many valuable suggestions. The author also had many useful discussion with his colleague Mr. R. Chebili.

The author is indebted to Mr. ALF.Prideaux, the chief technician, Mr. A.Clarke the technician of the Soil Mechanics laboratory and Mr. D.Skinner the technician of the Structures laboratory for their help during the experimental work.

Finally, the author would like to acknowledge the financial support from the government of the Arab Republic of EGYPT.

CONTENTS

NOTATION

A	cross section area
M	bending moment
P	load
P_E	Euler load
k	normal curvature
H	mean curvature
G	Guassian curvature
E	modulus of elastcity
ν	Poisson ratio
E_t	tangent modulus of elastcity
$\bar{E}$	reduced modulus of elastcity
I	second moment of area
N_x	normal force in x-direction
Ny	normal force in y-direction
N_{xy}	membrane shearing force
D	flexural rigidity
$\dot{\varepsilon},\sigma$	strain rate and stress
$\phi^{(i)},\phi^{(c)}$	instantaneous and creep deformation
M_c,M_n	constant bending moment
n,k	temperature constants
n	normal vector
t	tangent vector
b	binormal vector
i,j,k	unit vectors in x,y,z directions
g^1,g^2 and g^3	contravariant metric tensors

g_1,g_2 and g_3	covariant metric tensors
ds	distance between two adjacent points on a surface
du	deformed length
α,β	axial and bending stiffness
$\frac{d\psi}{du}$	curvature
$\left(\frac{ds}{du}\right)_i$	initial curve length.
$\left(\frac{ds}{du}\right)_f$	final curve length.
S	spring stiffness
δ	spring displacement
Q_1,Q_2,Q_3,Q_4	bending, axial, spring and loading terms of the strain energy equation
BI1,BI2,BI3,BI4,BI5	integration values of [B] up to the fifth integration
BD1,BD2,BD3,BD4	differentiation value of [B] up to the fourth derivative
P_{ij}	nodal load
M_B and M_g	bending moment about axis in plane of surface and bending moment about normal axis
M_τ	torsion moment,
k_B,k_G and τ	normal curvature, geodesic curvature and twist
Tj	tension forces in the surface.
Δ_1,Δ_2, ,Δ_n	control points coordinates in the strain energy equation
β_1	bending stiffness in x direction
β_2	bending stiffness in y direction

J	torsion factor
k_B	normal curvature
k_G	geodesic curvature
τ	twist
T_j	tension forces in surface
Q_1, Q_2	normal and geodesic bending terms in virtual work
Q_3	torsion bending term in virtual work equation
Q_4,	supported spring (boundries) term in virtual work
Q_5	external load term in virtual work equation
$\epsilon_{12}, \epsilon_{21}$	special skew symmetric tensors
b_{11}, b_{22}	symmetric surface tensors
$\Gamma^2_{11}, \Gamma^1_{22}$	Christoffel symbols of the second kind
x,y,z	coordinates of the control points

(A copy of these notations is inclosed in a folded sheet at the end of the thesis)

INTRODUCTION

Exhibition and sports centres need a structure that can cover large spans with the minimum number of intermediate supports. Such structures can be constructed using many different structural forms including trusses and spacframes, tents and cable nets, air supported structures and shell structures. This thesis will study the structural behaviour of one particular type of shell, the grid or lattice shell.

The concept of the grid shell was introduced in 1946 by Professor Frei Otto of Stuttgart University. He investigated the design of dome structures formed by inverting the shape of a hanging chain net in which self-weight produces direct forces only. These forces are tensile when the net is hanging and will change to compressive when the net inverted into a standing grid as shown in Fig.(1).

With this technique, Professor Otto was extending the method employed by the Spanish architect, Antoni Gaudi, at the end of the nineteenth century for the design of masonry structures. Methods of graphical analysis were popular with engineers at that time, and Gaudi developed their two-dimensional modeling into three dimensions.

The first grid shell structure was erected for the German Building Exhibition at Essen in 1962 (Roland Conrad (1975))* ,

References are listed by author in alphabetical order at the end of the thesis.

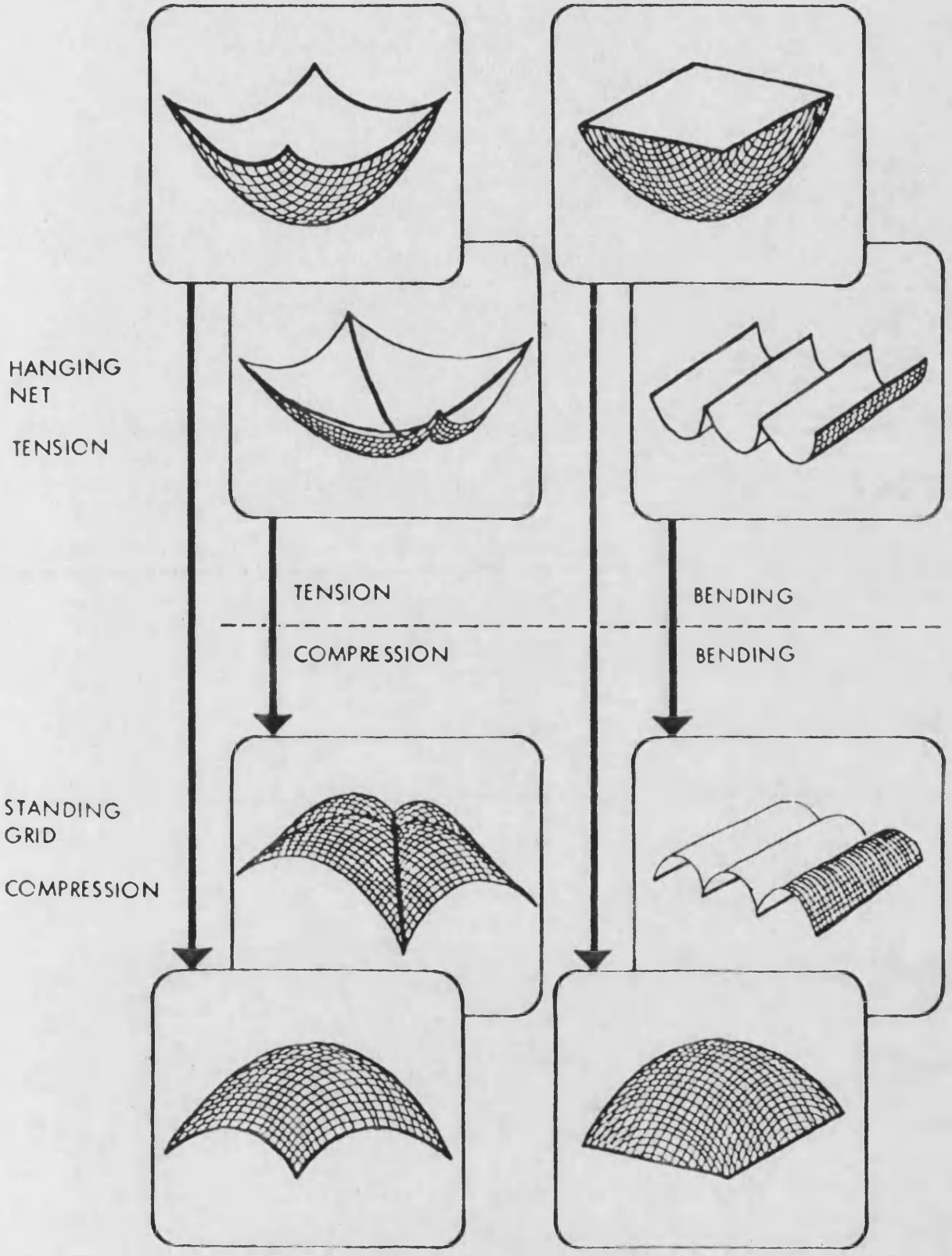


Fig.(1) Types of stresses (From IL-10 (1975))

by professor Otto. He developed an erection method in which the shape of a hanging uniform mesh chain net is recreated by a lattice of wooden rods. The lattice can be prefabricated flat as an equal grid, pulled or pushed up and then fixed to boundary supports. This erection technique requires relative rotation of the lattice members and therefore only single bolts can be used where the members cross and the bolts need to be loose during erection.

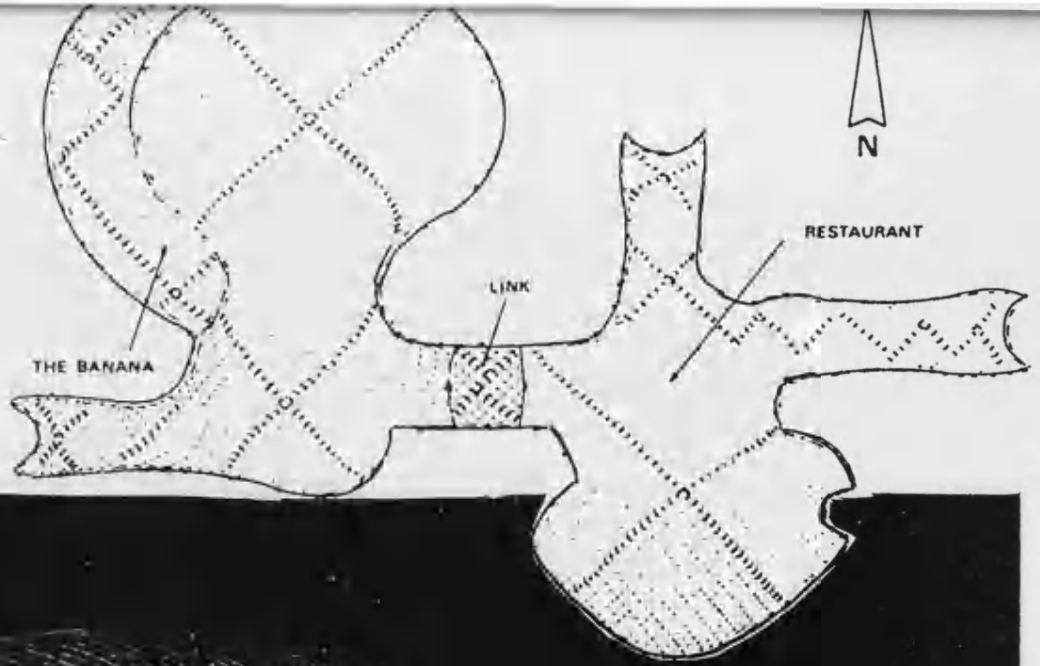
After erection the bolts are tightened and this provides a certain amount of 'in-plane shear stiffness'. If expected deformations are excessive, diagonal ties may need to be provided.

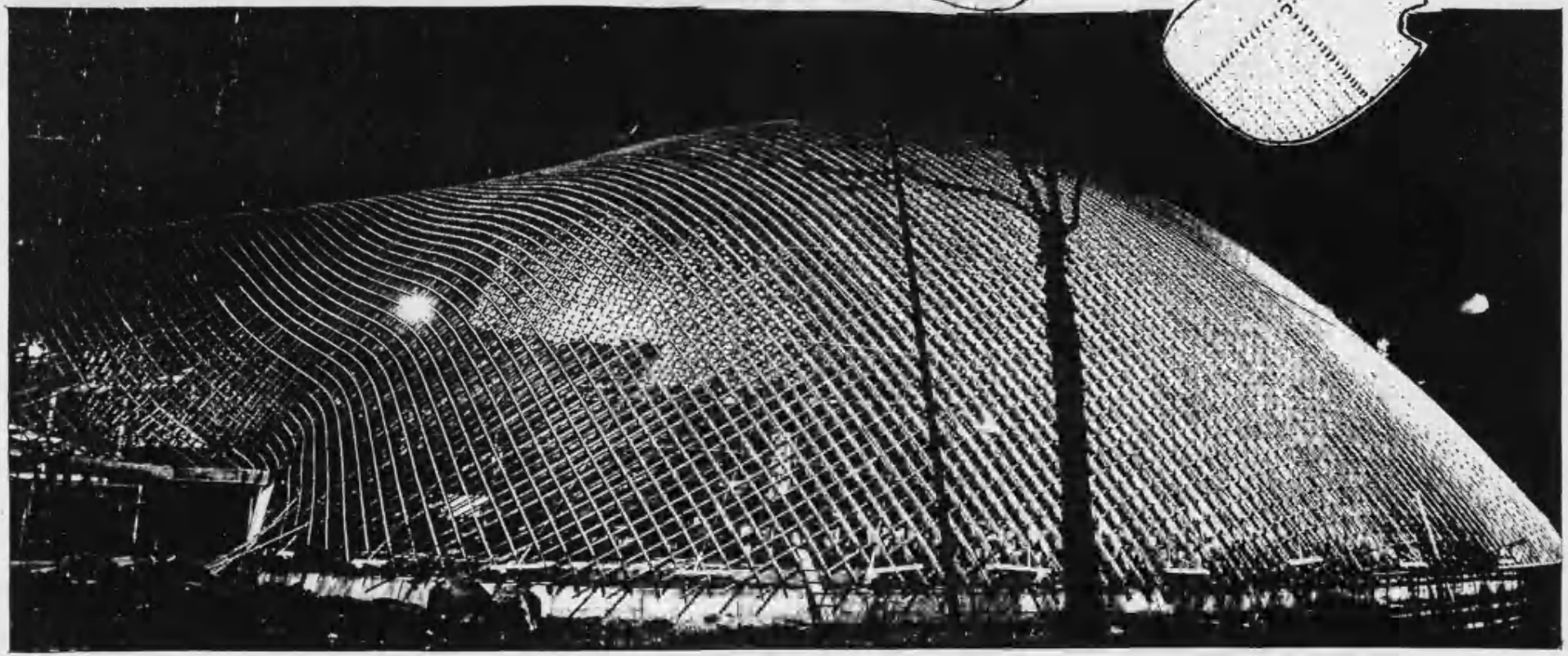
Following the Essen Exhibition Building, grid shells were constructed at the German Federal Pavilion for Expo.'67 at Montreal and the Mannheim Exhibition Centre (1974). The two Mannheim grid shells were made from double layer grids to cover an area 74066m^2,(see Fig.(2) from Happold and Liddell (1975)). A number of smaller grid shells as shown in IL-10 (1974) were also constructed.

However, grid shells are not popular with engineers because they are highly nonlinear in their structural behaviour and are prone to buckling and creep buckling. Also there is no clear and direct design method for this type of structure.

The main task of this thesis is to present the nonlinear analysis of grid shells and to build up a computer program to

Fig.(2) Mannheim Exhibition Centre
(From Happold and Liddell 1975)





calculate the buckling load of the grid shell.

This thesis is divided into two parts. Part I discusses the principles and analytical methods employed in buckling theory, the creep buckling, the finite element method, the differential geometry and the theory of spline functions.

Part II applies these methods to the numerical analysis of the buckling of grid shells. A finite element program initially in two-dimensions is described which uses cubic B-spline element. Then the program is extended for the grid shell which uses a new bicubic B-spline finite element.

The results of physical model tests are presented which show a good agreement with the prediction of the numerical analysis.

Readers who are familiar with the theory described on Part I may prefer to start reading Part II which refers back to Part I. It was thought necessary to include the contents of part I since very few engineers are familiar with the theories it describes.

PART I

CHAPTER ONE

BUCKLING OF STRUCTURES

1-1 Buckling characteristics of structures

Buckling leads to the failure of many structures, including shells, without any obvious warning. That is why one of the most main problems in the theory of elasticity is the prediction of the buckling load of thin-walled structures with either single or double curvature.

Buckling could be defined as the loss of strain stiffness due to compressive forces within a structure. Sometimes the structure remains elastic during buckling and in other cases buckling may be complicated by yielding and even brittle fracture.

Buckling leads to excessive deflection but the deflection prior to buckling may be small or large. Thus in the case of the arch shown in Fig.(1-1-a) subjected to a uniform load, deflection will be small until buckling occur.

However, the concentrated load case shown in Fig. (1-1-b) will undergo large deflection buckling. In practice, buckling is rarely a problem for arches subject to non-uniform load since the bending strength of the arch will be exceeded before the deflection become large.

(a) (b)

Fig.(1-1)

Most of the standard analytical work that has been done on the buckling of structures assumes small displacements prior to and during the initiation of buckling. This assumption produces a linear theory which also becomes as eigenvalue theory.

Simple model tests shows that linear theory is unlikely to be of much value when applied to grid shells. However it is necessary to have some understanding of linear buckling before going on to study the far more complicated subject of non-linear buckling.

The following section will introduce the ideas and methods used in buckling analysis, starting with the simplest case, pin-ended column.

1-1-1 Buckling of a uniform pin-ended elastic column.

In 1744 Euler investigated the deformation of an initially straight uniform elastic rod under endwise compression. There are

three conditions (Calladine (1983)), which must be satisfied simultaneously at every point on the deformed centre-line of the rod;

a) The equilibrium relation between the load, **P**, and the bending moment, **M**, in the rod.

b) The elasticity relationship (Hooke's law) between the bending moment and curvature k.

c) The compatibility relation between the profile of the displaced moment and local curvature.

It is assumed that the axial stiffness of the rod is sufficiently great for deformation to take place by bending only. The exact expression for the local curvature, k, of a plane curve was written by Euler as follows;

$$k = \frac{\dfrac{d^2y}{dx^2}}{\left[\,1 + \left(\dfrac{dy}{dx}\right)^2\,\right]^{3/2}} = \frac{M}{EI} = -\frac{Py}{EI} \qquad (1\text{-}1)$$

where E and I are the modulus of elasticity and second moment of area. This equation may be simplified, when $\left(\frac{dy}{dx}\right) \ll 1$ and by eliminating the moment, M, a differential equation for the unknown displacement y is given;

$$EI\,\frac{d^2y}{dx^2} + Py = 0\,. \qquad (1\text{-}2)$$

The solution of this equation is

$$y = A \sin \mu x + B \cos \mu x \qquad (1\text{-}3)$$

where A and B are constants of integration, which will be subject to boundary conditions at the ends of the column.

Let;

$$y=0 \quad \text{at} \quad x=0 \quad \text{and} \quad x=L \tag{1-4}$$

equation (1-3) gives,

$$B=0 \quad \text{and} \quad A \sin \mu L=0 \tag{1-5}$$

where; $\mu L=i\pi$ $(i=1,2,3, \ldots)$,

L is the column length

The second solution of the equation is satisfied for any value of A including zero. Therefore the shape of the column deflection is;

$$y=A \sin \mu x \quad \text{or} \quad y=A \sin \frac{i\pi x}{L} \tag{1-6}$$

this represents the buckling mode of the column as a sine wave of half-length (L/i) and of arbitrary amplitude A at a certain values of the load P. The lowest of these is known as the Euler buckling load,

$$P_E=\frac{\pi^2 EI}{L^2} \tag{1-7}$$

and the resulting stress $\sigma_E=\frac{\pi^2 Er^2}{L^2}$ (1-8)

where; r is the radius of gyration

Equation (1-8) shows the stress σ_E depends only on the elastic modulus of the material and the slenderness ratio L/r. The Euler buckling load can represent the column behaviour at L/r greater than 100. For values less than 100 there are theoretical

and empirical forms that should be taken into consideration (see Marshall 1969), and can be summarized as follows;

a) The reduced modulus theory

Consider a column subjected to an axial load P greater than P_E and another disturbing force applied to produce a slight deflection. This will increase the stress on the concave side and reduce it on the convex side of the buckled column. On the concave side the rate of increase will be proportional to $d\sigma/d\varepsilon = E_t$ where E_t is the tangent modulus at the stress σ. However, on the convex side the reduction relieves only the elastic portion of the strain and the normal value of E will be applied. Hence, the equation (1-2) of the column can be written as,

$$\frac{d^2y}{dx^2}(EI_1+E_tI_2)=-Py \qquad (1\text{-}9)$$

where I_1 and I_2 are the second moments of the areas to the left and right of the new neutral-axis.

Putting $\hat{E}I=EI_1+E_tI_2$

gives

$$\hat{E}I\frac{d^2y}{dx^2} + Py = 0 \qquad (1\text{-}10)$$

then the reduced modulus $\hat{E}$ can be calculated as;

$$\hat{E} = E\frac{I_1}{I} + E_t\frac{I_2}{I}. \qquad (1\text{-}11)$$

This is similar to the Euler equation and the value for the critical load P_r is

$$P_r = \frac{\pi^2 \hat{E} I}{L^2} \quad . \qquad (1\text{-}12)$$

b) The tangent modulus theory

The tangent modulus theory assumes that no strain reversal takes place and that the tangent modulus E_t applies over the whole cross-section. The analysis is then the same as the standard Euler approach with the substitution of E_t for E gives the critical load P_t (see Fig.(1-2));

$$P_t = \frac{\pi^2 E_t I}{L^2} \qquad (1\text{-}13)$$

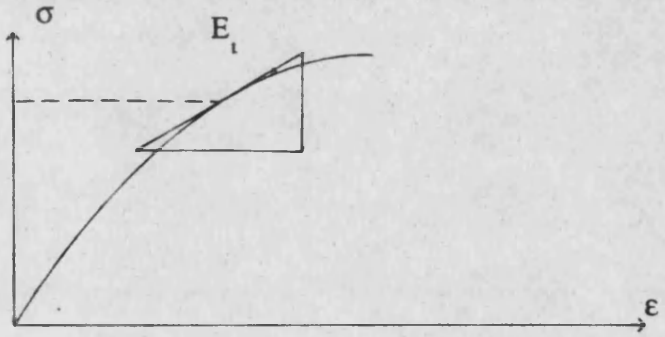


Fig.(1-2)

$\hat{E}$ is greater than E_t also the critical load from the reduced modulus approach is greater than that derived from the tangent modulus theory.

c) Shanley's column explanation

Shanley showed that there is a continuous spectrum of load increases from P_t to P_r with the deflection increasing from zero

at the former to infinity at the later. He shows that the relationship between the load $P(>P_t)$ and the deflection y is given by;

$$P = P_t\left(1+ \frac{1}{b/2y+(1+\tau)/(1-\tau)}\right) \qquad (1\text{-}14)$$

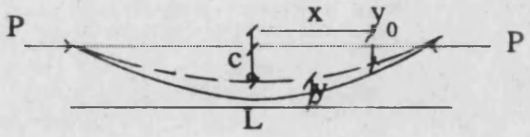
where;

$\tau=E_t/E$ and is assumed constant,

b is column diameter.

d) Perry-Rebertson Formula

The perry-Rebertson approach is based on the assumption that the column has an initial curvature which is given by the equation



$$y_o = c_o \cos \pi x/L \qquad (1\text{-}15)$$

where c_o is the initial departure from straightness at the centre of the column. Under a load P the deflection is increased by an amount y, the differential equation of bending being;

$$EI \frac{d^2y}{dx^2} = -P\ [y+c_o\cos(\pi x/L)] \qquad (1\text{-}16)$$

or

$$\frac{d^2y}{dx^2} +\mu^2[(y+ c_o \cos(\pi x/L)]=0 \qquad (1\text{-}17)$$

where $\mu^2=P/EI$. A solution of this equation, similar to equation (1-1), is shown by back substitution to be

$$y = A \sin \mu x + B \cos \mu x + \frac{\mu^2 c_o \cos(\pi x/L)}{\pi^2/L^2 - \mu^2} . \qquad (1\text{-}18)$$

In order to discuss what happens as the deflection gets large after buckling has started, the results can be summarized as shown in Fig.(1-3).

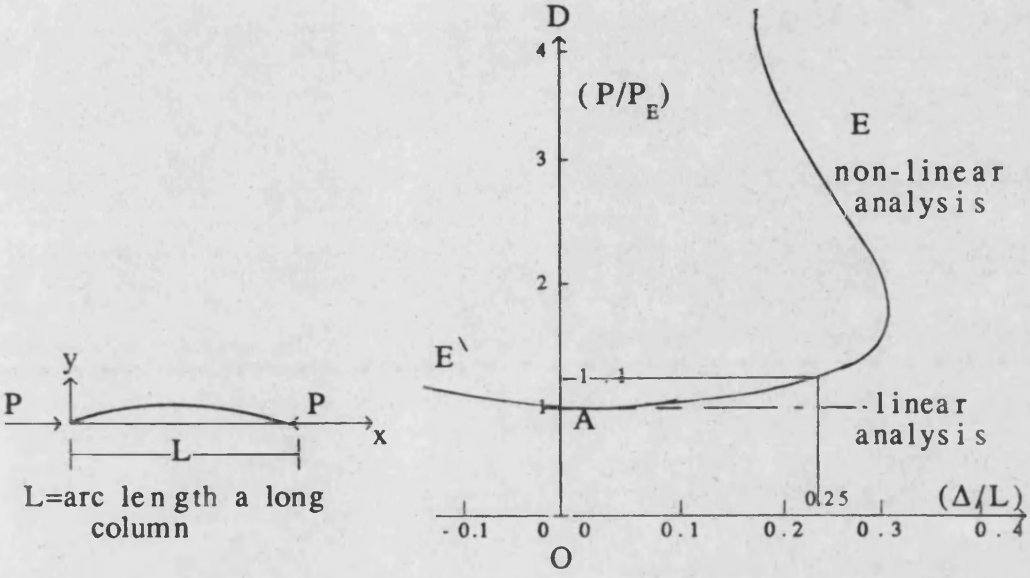


Fig.(1-3) A uniform pin-ended elastic column subjected to a compressive load

The load-displacement relationship is plotted in non-dimensional form, the horizontal axis is the lateral displacement, Δ, (measured at the midpoint of the column) divided by L the arc length of the column. The vertical axis is the load, P, divided by the Euler load. When P/P_E is less than 1, the column is in equilibrium only when it is straight; this state, which is one of the stable equilibrium, is represented by the line OA.

When P/P_E is equal to or greater than 1, there are three possible equilibrium positions. The first, represented by the line AD, implies that the column remains straight; however, this is a position of unstable equilibrium which cannot be sustained in practice without artificial assistance. The second equilibrium position is stable, and it is represented by the curve AE, the figure shows that the value of P rises only to $1.1P_E$ when $\Delta/L=0.25$ which is certainly a 'large deflection' for most of the problems. The third equilibrium position is merely a reflection of the curve AE (shown as AE\` in the diagram), including the alternative possibility of displacement in the opposite direction.

The equilibrium path OA is said to bifurcate at point A (where the end force reaches its critical value, P_E) because it forks there and presents a choice of stable paths (AE, AE\`) and an unstable path (AD), of which one of the stable paths will normally be followed.

All of these predictions are valid only if the column remains perfectly elastic. In reality it is to be expected that a load will eventually be reached at which, somewhere in the column, the material reaches its limit of proportionality. Further increase in the load leads to the development of a zone in which the material is inelastic. In this condition the column is less stiff than it would have been had if it remained elastic. Only when the column is relatively short and thick the inelastic behaviour is likely to happen early.

Provided that, the column remains elastic the load can increase and buckling is initiated at the eigenvalue load, P_E. This case becomes as the stable post buckling and it means that the eigenvalue load is conservative and therefore safe.

However many structures, particularly shell structures exhibit unstable post buckling in which the load drops off rapidly after buckling. In practice this means that imperfection of the structure makes it impossible for the structure to resist the eigenvalue load.

In fact, it can be shown that the two-column structure shown in Fig. (1-4) exhibits unstable post-buckling. This is because the column which buckles first attracts more than its 'fair share' of the load.

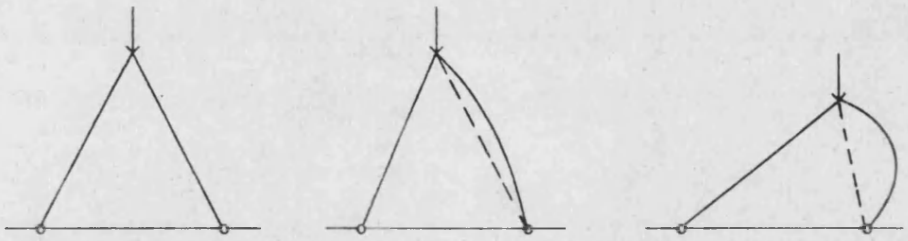


Fig.(1-4)

The initial imperfection as well as the creep buckling, in general, lead to a considerable reduction in the maximum load which the column can carry, (the effect of creep buckling will be studied in more detail in chapter 2).

The behaviour of the initially imperfect structure is best understood, as explained by Allen and Bulson, by referring to the inset diagram (1-5). As the end load is increased from zero, the stable curve AB is followed, then displacements increase more rapidly than the load. When the maximum is reached at B, and provided that the load is held constant, the structure collapses at point E on the stable path DEF, along which the structure is in tension.

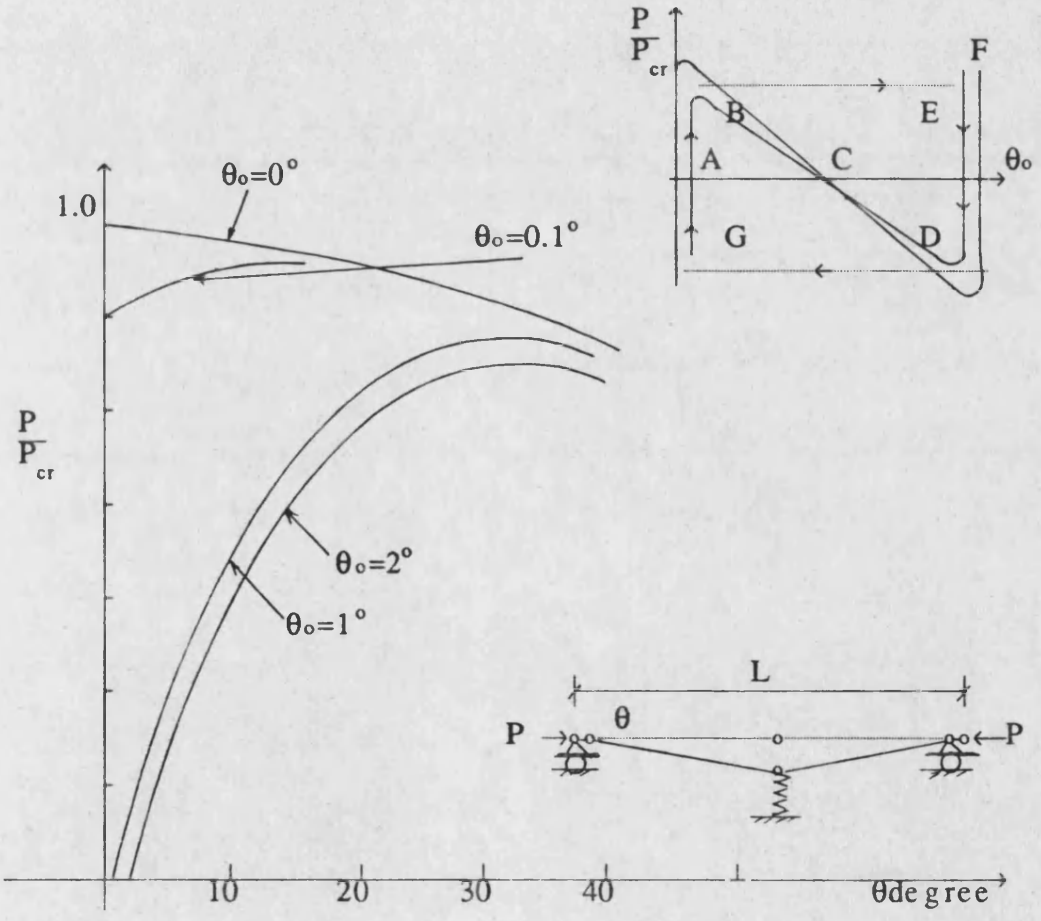


Fig.(1-5) Force-displacement curves for a light rod-and-spring structure

If the load is then progressively reduced in magnitude, the path ED is followed and the structure eventually collapses again along DG. The unstable path BCD can be followed only if the structure is held in a deflected state while the load is allowed to adjust itself. It is very important to recognize that the sensitivity of the maximum load to the size of the initial imperfection, which is a characteristic of nonlinear unstiffening structures, cannot be predicted by any ordinary small-deflection analysis. Fig.(1-6) shows the maximum axial force against the initial imperfection θ_o.

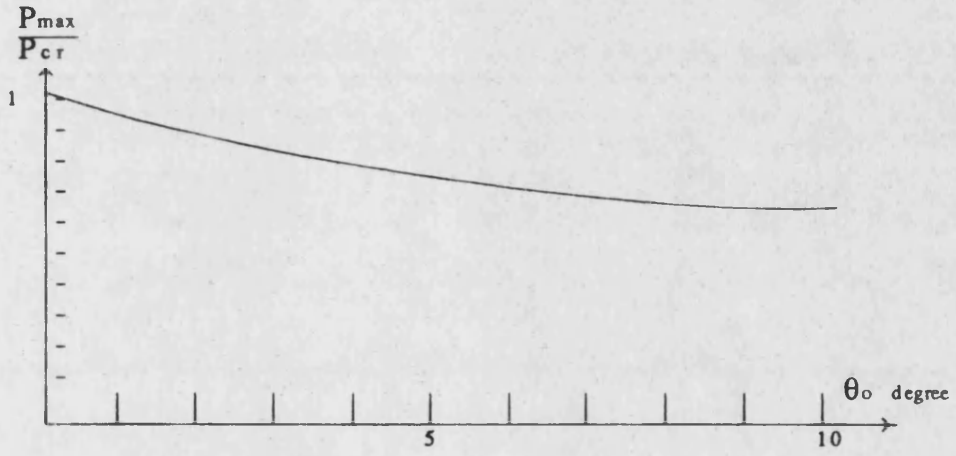


Fig.(1-6) The relation between the maximum load and the initial imperfection

1-1-2 Linear buckling of plates

The theory of bending of thin plates is similar to the theory of beams, but the effect of bending in two directions has to be taken into account. When the plate deflects, due to the

application of an external load, it will bend to radii of curvature in two directions ρ_x ,ρ_y. The early treatments of the bending theory was made by Navier in 1820 based on the following assumptions;

1) Deflections are small (less than the thickness of the plate).

2) The middle plane of the plate does not stretch during bending, and remains a neutral surface.

3) The thickness of the plate is small compared with other dimensions.

4) Plane sections rotate during bending to remain normal to the neutral surface and do not distort, so that the stresses and strains are proportional to their distance from the neutral surface.

The basic bending equation of the plate can be written as follows (see Timoshenko (1963),(1970) and Allen (1980));

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{q}{D} \qquad (1\text{-}19)$$

where q is the intensity of the distributed lateral load,

and $D = \frac{Et^3}{12(1-\nu^2)}$ is the flexural rigidity of the plate element,

ν Poisson ratio.

Equation (1-19) is analogous to the basic equation for the bending of a beam under lateral load ($\partial^4 w/\partial x^4 = P/EI$). If, in addition to the lateral load, there are forces applied at the edges of the plate, acting in the middle plane such as; N_x and N_y per unit length applied normally to the edges and N_{xy} per unit

length as shearing forces as shown in Fig.(1-7), the general equation for the buckled plate can be summarized as follows;

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{1}{D}\left(q + N_x \frac{\partial^2 w}{\partial x^2} + 2N_{xy} \frac{\partial^2 w}{\partial x\, \partial y} + N_y \frac{\partial^2 w}{\partial y^2}\right) \quad (1\text{-}20)$$

It should be emphasised that equation (1-19) for plate is the equivalent of equation (1-2) applied to the column and the linear equation (1-20), only applies to small deflections. Assuming that there is no lateral load (q=0), in equation (1-20) then;

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{1}{D}\left(N_x \frac{\partial^2 w}{\partial x^2} + 2N_{xy} \frac{\partial^2 w}{\partial x\, \partial y} + N_y \frac{\partial^2 w}{\partial y^2}\right) \quad (1\text{-}21)$$

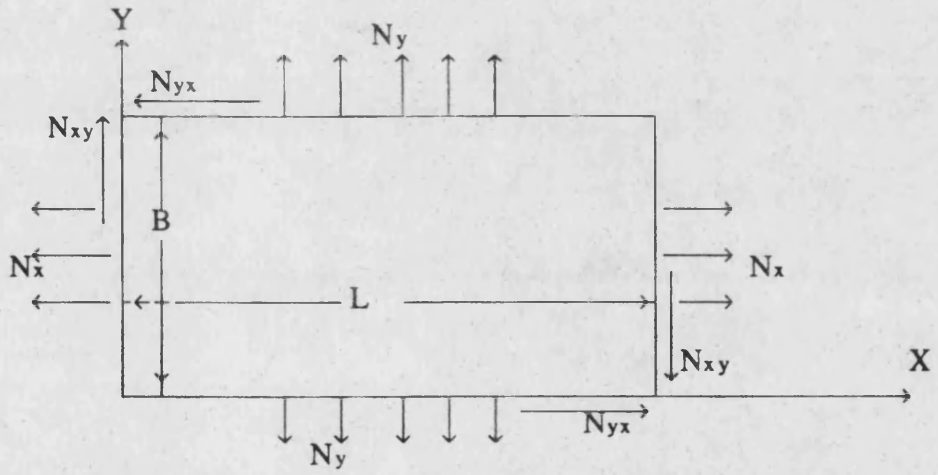


Fig.(1-7) Notation for the buckling of a simply-supported uniform rectangular plate

The above equation is the basic differential equation of plate buckling and by using the Ritz equation (as in Bleich (1952)) or the energy equation, the solution can be introduced.

However, the expression of the strain energy stored in a complete plate derived by Bryan 1891 is,

$$U=\frac{D}{2}\iint\left\{\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}\right)^2-2(1-\nu)\left[\frac{\partial^2 w}{\partial x^2}\cdot\frac{\partial^2 w}{\partial y^2}-\left(\frac{\partial^2 w}{\partial x\,\partial y}\right)^2\right]\right\}dxdy \quad (1\text{-}22)$$

Thus this equation can give neither information about the post-buckled stability on the effect of imperfection nor cover the problems of large deflections.

Since the surface was originally flat, it can only have curvature and twist on account of normal displacement w. There is almost certainly in general, a change of Gaussian curvature when the initially flat surface adopts a buckled configuration; consequently there must necessarily be some nonzero surface strains which are required, by Hooke's law, corresponding to changes in N_x, N_y and N_{xy}. A check may be done on the magnitude of these changes for any postulated modeform in comparison with the values of the stress resultants which are applied at the edges. The local change of Gaussian curvature is proportional to the square of the amplitude of w. This explains the need for the assumption that w is small. The relationship between Gaussian curvature and surface strain is discussed in chapter 4.

The behaviour of plates containing small initial deviations from flatness is similar to that for columns. With a lateral deflection of amplitude (et), the deflection grows from the onset of loading and there is no load at which buckling can be said to occur. As it can be seen from Fig.(1-8), the effect of

imperfection is most noticeable in the region of the flat plate buckling-load and becomes small remote from it.

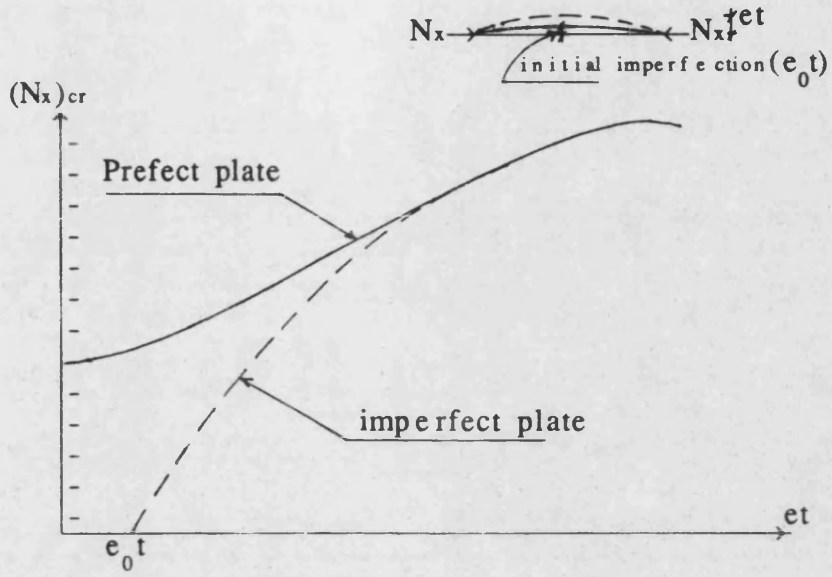


Fig.(1-8) Deflection of plates with initial imperfection

1-1-3 Linear buckling of shells

In deriving the expressions for the strain energy equation of thin shells, both bending and stretching strains must be taken into account. Consider first the small shell element drawn in Fig.(1-9) formed by the intersection of two pairs of adjacent planes normal to the middle surface. The radii of principal curvatures are r_x and r_y in the xz and yz planes respectively. When the shell element deflects due to application of external load, the radii of curvature will change to $r_x^{\backslash}$ and $r_y^{\backslash}$ then the change in curvature are ρ_x,ρ_y and ρ_{xy}.

The strain energy due to bending with no middle surface stretching can be derived from the expression of flat plates from equation (1-22) as follows;

$$U_b = \frac{D}{2} \iint \left[(\rho_x + \rho_y)^2 - 2(1-\nu)(\rho_x \rho_y - \rho_{xy}^2) \right] dA \qquad (1\text{-}23)$$

and the strain energy due to stretching of the surface is;

$$U_s = \frac{Et}{2(1-\nu^2)} \iint \left[(\varepsilon_1+\varepsilon_2)^2 - 2(1-\nu)\left(\varepsilon_1 \varepsilon_2 - \frac{\gamma_{xy}^2}{4} \right) \right] dA \ . \qquad (1\text{-}24)$$

where;

$\varepsilon_1, \varepsilon_2$ are the middle surface strain in x and y direction respectively,

γ_{xy} is the component of shear strain in the middle surface of the shell,

dA,t are the infinitesimal area and the element thickness.

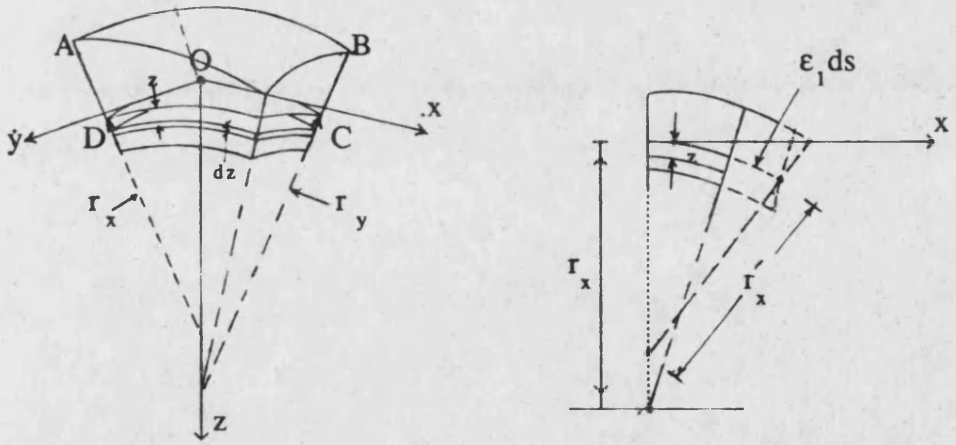


Fig. (1-9) shell element

A numerical study has been carried out by D.Wright (1965) on reticulated or framed shells. This study covered some relationships between the shell membrane forces and bar forces,

elastic properties of the analogous shell, and the buckling criteria of individual members, for 'snapping' and for the overall shell instability.

Most of the cylindrical shells have small initial deviations, Donnell and Wan (1950) show that the smallest imperfection leads to a considerable reduction in the peak compressive load as shown in Fig.(1-10) where σ_{min} and σ_{max} are the minimum and maximum stresses, and σ_{cr} and ε_{cr} are the critical values of stress and strain respectively.

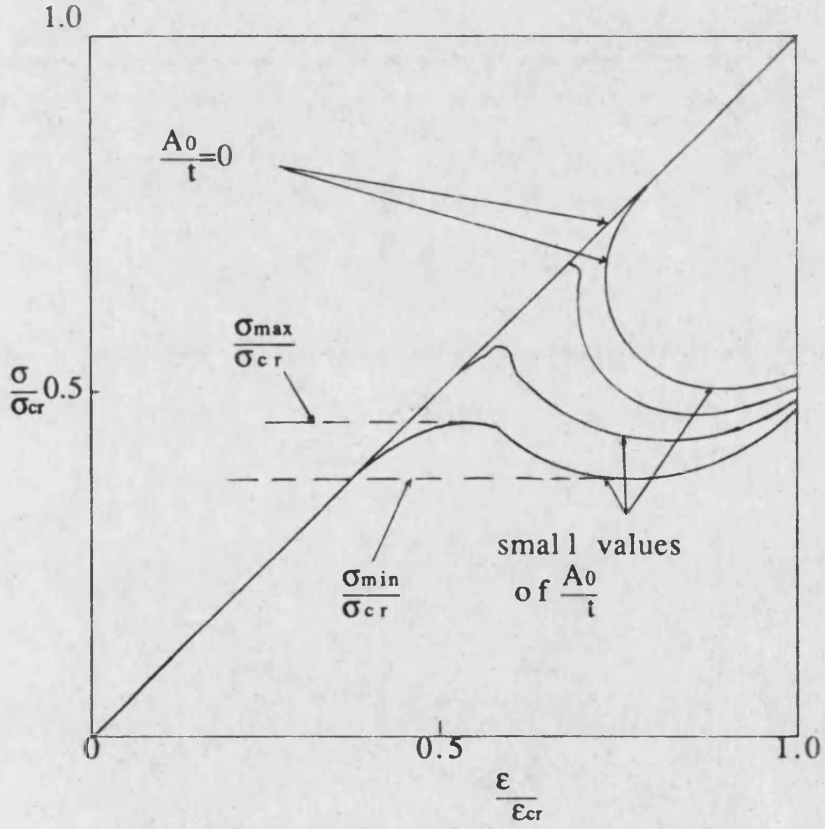


Fig.(1-10) Imperfection in cylindrical shells

1-2 Matrix formulation

Problems in structural analysis usually lead to simultaneous equations and/or differential equations which need to be solved. Analytical solution to a differential equation can usually be found in simple special cases (often only after assuming linear behaviour). The finite element and finite difference methods, for instance, are based on the role of replacing a differential equation by a large number of simultaneous equations which may be linear or non-linear.

1-3 Linear behaviour

The differential equation governing deflection of a pin-ended column is,

$$M=P_{cr}.y = -EI \frac{d^2y}{dx^2} \qquad (1\text{-}25)$$

the buckling load can be obtained by using the finite difference method therefore, equation (1-25) can be written as;

$$[A]_{nxn}\{X\}_{nx1}=\mu[B]_{nxn}[C]_{nxn}\{X\}_{nx1} \qquad (1\text{-}26)$$

where;{X}is a square matrix of all the eigenvectors **x**,

μ is the load factor

[C] is a diagonal matrix contain the stiffness terms;

[A] & [B] are traditional matrices .

Multiplying both sides by $[A]^{-1}$ then the equation will be;

$$[H]\{X\} = \gamma\{X\} \tag{1-27}$$

where;

$$[H]=[A]^{-1}[B][C] \tag{1-28}$$

$$\gamma = \frac{1}{\mu} .$$

Where the solution of equation (1-28) is an eigenvalue problem.

If the matrix H is symmetric and positive-definite, all the eigenvalues are positive and real, and n eigenvalues exist. On the other hand, the structure becomes unstable when its stiffness matrix becomes singular or the determinant would become zero.

Similarly in the case of plate buckling when an in-plane load is applied, the governing differential equation can be written as;

$$D\left[\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\right] + \left(P_x \frac{\partial^2 w}{\partial x^2} + 2P_{xy} \frac{\partial^2 w}{\partial x\, \partial y} + P_y \frac{\partial^2 w}{\partial y^2}\right) = q \tag{1-29}$$

Substituting the external applied forces by $\mu\{P\}$ where μ is constant corresponds to the critical load and putting the above equation in a matrix form, then equation (1-29) can take this form;

$$[K]\{w\}=\mu[A]\{w\} \tag{1-30}$$

where, [K] represents the left hand side of the equation (1-29)

and [A] represents the values $\frac{\partial^2 w}{\partial x^2}$, $\frac{2\partial^2 w}{\partial x\, \partial y}$, $\frac{\partial^2 w}{\partial y^2}$ multiplied by the known P-values. Multiplying both sides of equation (1-30) by $[K]^{-1}$ then,

$$[G]\{w\}=\gamma\{w\} \tag{1-31}$$

where, $[G]=[K]^{-1}[A]$ and $\gamma=\frac{1}{\mu}$

The solution of the equation (1-31) for the largest eigenvalue γ gives the smallest value of μ which corresponds to the critical load and the eigenvector **w** is the deflection.

Therefore, the solution of the linear buckling problem of pin-ended column or thin plates can be achieved by following the procedure of the 'eigenvalue' problem. The only requirement is that the equation should be in a certain matrix form, in another words the 'eigenvalues buckling' should be determined as may be explained below;

By assuming a system of n linear homogeneous equations with n unknowns as shown in matrix A, then there is at least one vector **x** and one corresponding scalar λ for which

$$[A]_{nxn}\{x\}_{nx1}=\lambda\{x\}_{nx1} \tag{1-32}$$

where **x** and λ are termed respectively an eigenvector and eigenvalue of matrix A. By introducing the unit matrix I, equation (1-32) could be rewritten as,

$$(A-\lambda I)x=0 \tag{1-33}$$

this equation may be written in full matrix as:

$$\begin{bmatrix} (a_{11}-\lambda) & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & (a_{22}-\lambda) & a_{23} & & \vdots \\ & & \cdot & & \vdots \\ a_{n1} & \cdots & \cdots & \cdots & (a_{nn}-\lambda) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = 0 \qquad (1\text{-}34)$$

where a_{11},a_{12} ,etc.., are the individual terms of A, and x_1, x_2,etc.., are the various unknowns. Equation (1-34) forms a homogeneous set of linear equations, whose determinant must be zero, i.e..

$$|A - \lambda I| = 0 \qquad (1\text{-}35)$$

This determinant could be expanded, providing an *n*th order polynomial in eigenvalue λ of the form;

$$\lambda^n + C_1 \lambda^{n-1} + C_2 \lambda^{n-2} + \qquad + C_n = 0 \qquad (1\text{-}36)$$

and the solution of this equation gives the eigenvalues λ_n which satisfy the conditions,

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} A_{ii} \qquad (1\text{-}37)$$

and

$$\lambda_1 \times \lambda_2 \times \cdots \times \lambda_2 = |A| \qquad (1\text{-}38)$$

Instead of evaluating the determinant, an iteration method could be used which leads to the eigenvalue with the largest absolute value.

1-4 Non-linear behaviour

The material nonlinearity has been heavily investigated by many authors such as Merchant (1954), Horne (1963), Tezcan (1966), (1969) and Majid (1970) in many applications of framework. Also, the geometric nonlinearity which is the situation in shells has been studied, (see Calladine (1977),(1986) and Haisler (1972)).

The solution of a nonlinear problem has been reduced to that of tracing a nonlinear load-displacement path by solving a system of nonlinear algebraic or differential equations. This solution is fully dependent on many factors which can be summarized as follows;

- The type of the problem to be solved.
- The degree of nonlinearity involved and the accuracy desired.
- The computer time required for a solution.

Assume the simultaneous equations of a nonlinear buckling problem in the following shape;

$$f_1(\delta_1,\delta_2,\delta_3, \quad \delta_n) = Q_1$$

$$f_n(\delta_1,\delta_2,\delta_3, \quad \delta_n) = Q_n$$

The majority of the matrix form of these equations can be written;

$$\{\delta^{i+1}\} = \{\delta^i\} + [\alpha^i][K^i]^{-1}.([\beta^i]\{R^i\}-\{C^i\})$$

where;

$\{\delta^i\}$ is the vector of nodal displacements at iteration i

$[K^i]$ is a matrix accounting for elastic and geometric stiff.

$\{R^i\}$ is the vector of residual nodal forces

$[\alpha^i]$ and $[\beta^i]$ are diagonal correction matrices and $\{C^i\}$ is a correction vector which may depend on previous residuals and projected stresses.

The calculation of the nonlinear differential equations can be obtained by using the incremental stiffness procedure (see Conner (1968) and Brebbia (1969)) for slight nonlinearity problem. Or by matrix iterative methods such as Newton-raphson or modified Newton-Raphson for highly non-linear problems ,(see Stricklin (1971), Mallet (1968), Haisler (1972) and Barnes (1984)).

1-4-1 Newton–Raphson method

The Newton-Raphson method is the most widely stable iterative analysis used for geometrically non-linear problems. In this method $[K^i]$ is the current tangent stiffness matrix, $\{C\}$ is null and, unless abnormal deflection increments or residuals are encountered, $\{\alpha\}$ and $\{\beta\}$ are identity matrices. More rapid convergence correction factor is,

$$\alpha_j^i = \frac{k_{jj}^{i+1} + k_{ij}^i}{2K} \qquad (1\text{-}39)$$

for the j^{th} degree of freedom, where the diagonal term k_{jj}^{i+1} is computed after determining the deflections $\{\delta^i\}$ using the tangent stiffness at iteration i. In other cases, when very low stiffnesses occur, it may be necessary to specify a maximum permissible deflection increment for any degree of freedom and

scale down all increments if this maximum is exceeded. Diagrammatically the procedure is represented in Fig.(1-11).

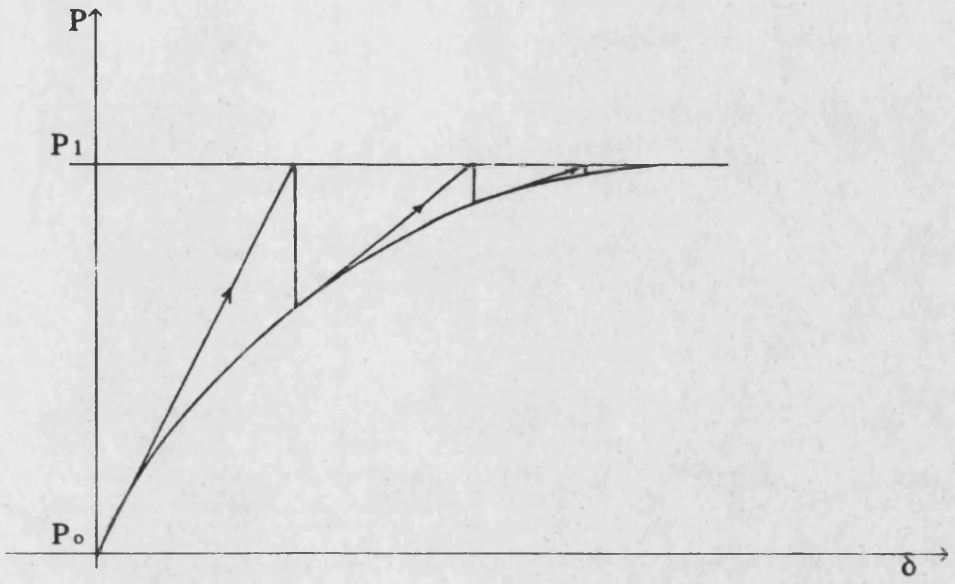


Fig.(1-11) Newton-Raphson method

1-4-2 Modified Newton-Raphson

A drawback of he Newton-Raphson method is that the tangent stiffness matrix has to be reset and solved at each stage. For this reason a "Modified Newton-Raphson" method is considered to be more efficient if the stiffness is held constant throughout. An alternative to using the initial value of the stiffness matrix throughout the process is to reset at intervals with a number of constant stiffness iterations within each interval. Although convergence is less rapid than for the standard Newton-Raphson method, computationally the process may be more efficient provided the system is not grossly non-linear. For a stiffening system,

however if the initial out-of-balance is too large, the analysis may diverge. Fig.(1-12) shows the procedure of the method.

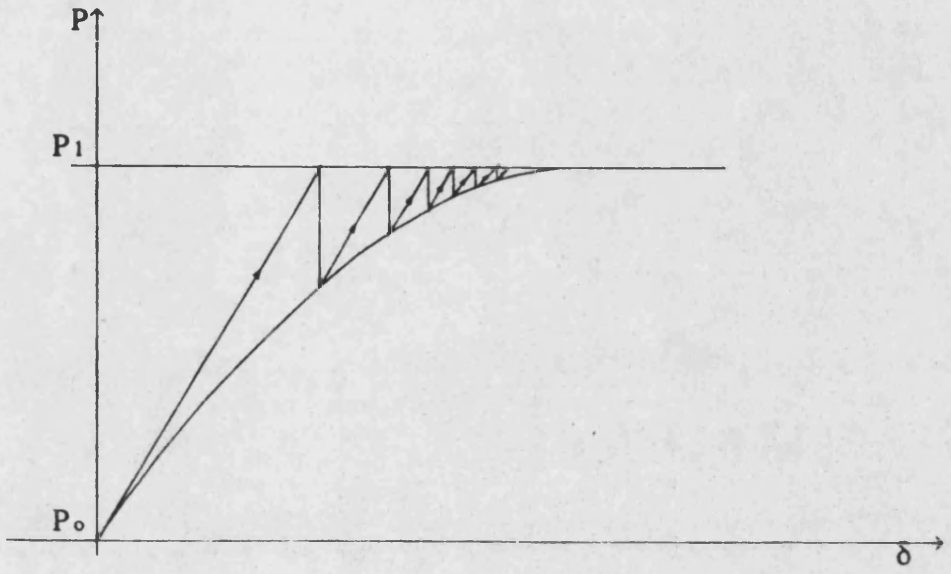


Fig.(1-12) Modified Newton-Raphson method

The solution of non-linear problems using the iterative analysis, usually, needs a number of cycles in order to achieve a certain accuracy as it will be explained.

1-5 Procedure of the iteration methods

The iteration procedure to determine the buckling load of the structures in general ,taking into account the non-linear behaviour, follows certain steps. The basic idea in this procedure is to perform a standard linear analysis under the action of a given set of external loads and then calculate the member end forces using the deformed geometry. If the member end

forces at a joint are not in equilibrium with the given external forces, the out-of-balance forces are applied on to the deformed geometry to yield another set of deformations and forces. If the new forces do not satisfy the joint equilibrium, the linear analysis continues with the latest geometry and with latest out-of-balance forces.

This procedure is repeated until equilibrium is reached at every joint. The original external loads are gradually increased and the equilibrium status is established at each time by following the process described above.

The magnitude of the external loads causing divergence in the unbalanced forces, in other words, producing excessive deformations at the joints, is considered as the buckling load of the system.

Calculating the buckling load in grid shells more or less follows the same process. The only difference is the out-of -balance forces are checked by making sure that the strain value in every patch in the shell, which is created as a result of forming the shell to a certain shape from the initial one, is zero. That will lead to modify the total stiffness matrix of the shell to include the part concerning the non-linear behaviour.

1-6 Buckling of grid shells

Grid shells could be considered as a system of interconnected arches and at the same time they behave in some ways like 'true' shells. Therefore the buckling behaviour of grid shells is a combination of the behaviour of arches and shells.

The understanding of the buckling of grid shells can be made easier by comparing the behaviour of a squash ball and a ping-pong (table tennis) ball when they resist a load. The squash ball represents a thick shell with comparatively high bending stiffness compared with membrane stiffness, while the ping-pong ball represents a thin shell. The crossing members of grid shells are free to rotate, so that their membrane stiffness is comparatively low meaning that they will behave as a squash ball.

A squash ball undergoes large deformation before the maximum load is reached, but a ping pong ball buckles suddenly after very little deformation. This applies even if holes are made in the balls to remove any effect of internal air pressure.

Using ties with grid shells may increase their shear stiffness and change their behaviour from thick to thin shells.

The main advantage of using ties on a grid shell is to increase the collapse load of the shell and control the large deflections. But on the other hand the shell becomes more sensitive to imperfections and creep which leads to a gradual

increase in the imperfections.

Ties were used on the Mannheim Exhibition Centre 1973. In model tests reported by Happold and Liddell (1975), the addition of ties increased the collapse load and reduced the size of the buckle. Also with ties the shell under uniform loading became more brittle and the collapse was sudden.

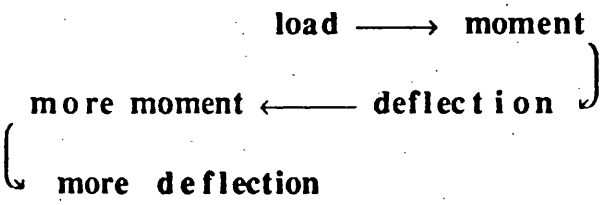
From this discussion, the buckling analysis of grid shells must be non-linear.

CHAPTER TWO

CREEP BUCKLING

2-1 General

The prediction of the buckling loads has been studied in chapter 1, and as mentioned, the creep buckling and the imperfection have a great effect on the exact values of the buckling load of the grid shell especially those made from timber. This effect can be considered as 'Vicious circle' for the behaviour of any structure under creep. When a load is applied, a moment and deflection will be created. This deflection will increase by the time and that means extra moment, thus leading to more deflection and this operation will be repeated until the structure collapse.



However, the effect of the creep buckling ,on different types of columns, will be discussed and recommended points will be explained in the case of grid shells.

2-2 Introduction and literature review

When a piece of material is subjected to external forces, it will deform elastically. If the load is increased, part of the

deformation will become plastic, that is will not be regained upon removing the load. When unloaded, some metals show an instantaneous reverse deformation followed by a slow recovery of part of the remaining deformation. Such time dependent deformation is termed creep, Fig.(2-1) shows the typical curve of creep and creep recovery.

There is always some eccentricity in all real columns. Therefore when an axial load is applied there will always be an increase in lateral displacement due to elastic, plastic and/or creep strains. The deflection due to creep will increase with time and this is equivalent to an increase in the initial imperfection. Catastrophic failure is a short term phenomena and occurs due to elastic instability, plastic deformation and/or brittle fracture.

There are three periods of time that can be observed during the creep rate process. As shown in Fig.(2-2), the first stage which is called the primary stage, usually occurs immediately upon the application of the load. the creep rate at that time is usually very high and the instantaneous deformation in this period may be partly elastic and partly plastic. Subsequently the rate at which the deformations proceed becomes constant; this is the secondary or the steady stage of creep and is usually the most important one in engineering applications. It is followed by a third period in which the creep rate is increasing. Hoff (1958) illustrated the three stages of strain and strain rate against the time.

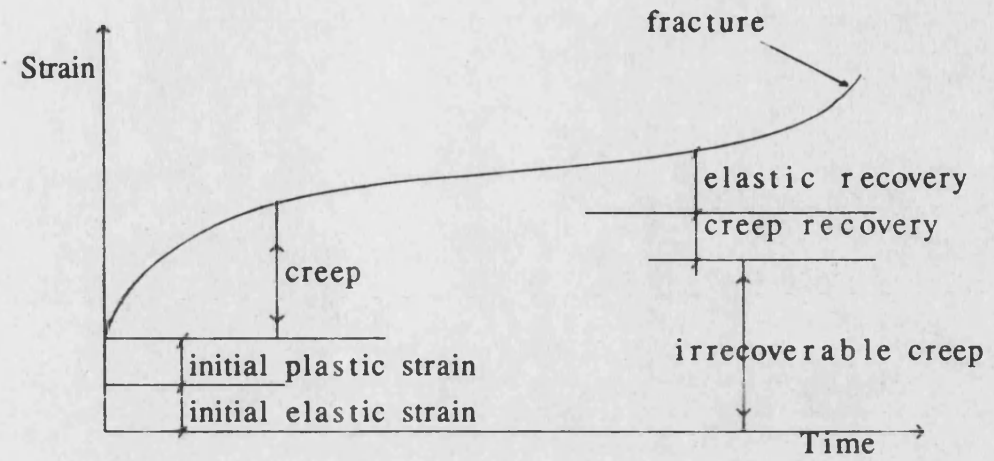


Fig.(2-1) Schematic curve of creep and creep recovery

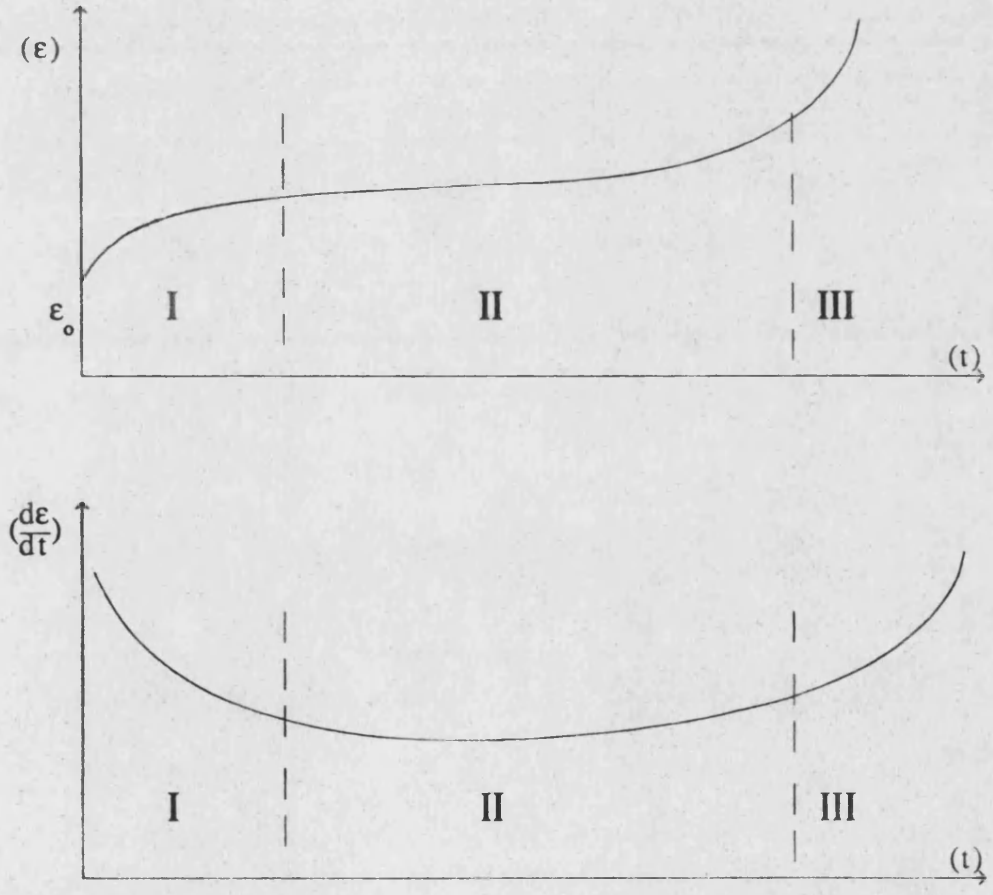


Fig.(2-2) Strain and strain-rate in constant stress creep test

The creep phenomenon is strongly temperature-dependent, the creep rates increase markedly with raising the temperature. On the other hand, creep buckling is a phenomenon which does not intrinsic dependently on temperature. Creep buckling has an interesting consequence for elements subjected to compression. As shown in Fig.(2-3) the column is slightly curved in its natural state before the compressive load (P) is applied to it. This is a reasonable assumption because all structures must have an initial imperfection, but never to zero tolerance. Similarly, an experienced testing machine operator can align the column a great deal of accuracy, but never perfectly.

In creep buckling the column load can not be based on a fixed critical load but rather depends on the allowable deflection which occurs to it at a certain time, Hult (1966) , Finnie, et al (1959) and Illston (1978). The first analysis of creep buckling appears to be due to Siegfried (1943), who pointed out that creep buckling could occur, and illustrated this for a material in which the strain rate is proportional to stress. The first reference to a specific problem of creep buckling appeared a few years later when Ross (1946) and (1947) studied the influence of creep on the buckling load of concrete columns.

From 1947 to 1957, many studies have been carried out to investigate the creep buckling of columns and plates followed by an interesting applications on bars and columns. In 1952 ,Libove, has studied a slightly crooked H-section columns for some materials. The material is characterized by a strain-time

relationship under constant uniaxial stress, which fitted the following expression;

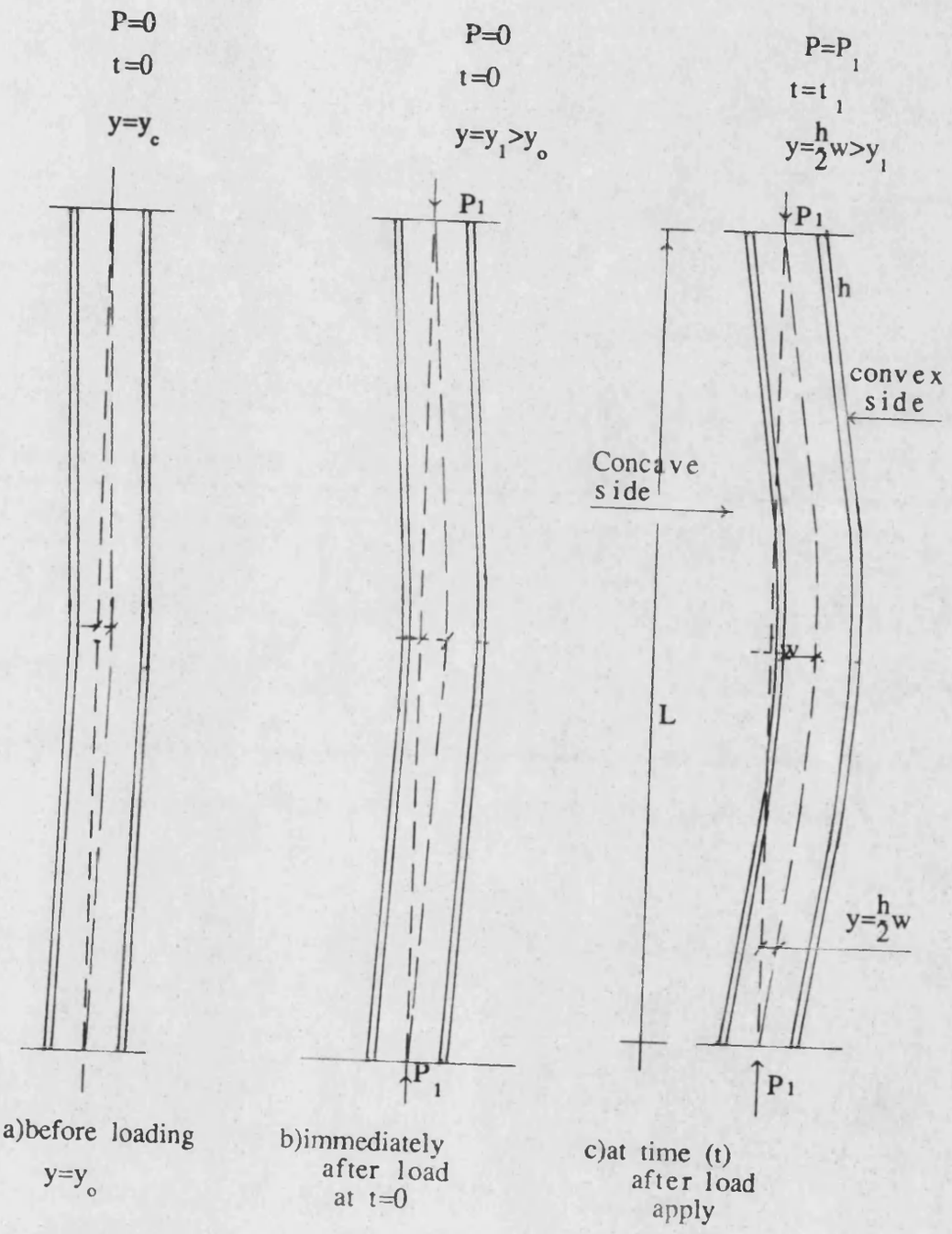


Fig.(2-3)

$$\varepsilon = \frac{\sigma}{E} + A\, e^{\beta\sigma}\, t^{k} \qquad (2\text{-}1)$$

where:- E,A,B and k are the material constants

ε, σ and t are the strain, stress and time subsequent to load application respectively,

Also the obtained strain increments are based on strain hardening.

The analysis assumes that the deflected shape of the column remains a sine-wave. This assumption enables the calculations to be based on only the centre cross section of the column and enables the results for the "H" column to be presented in a closed form. The accuracy of the method will depend on the agreement shown between the data and the previous empirical expression, which neglects initial plastic strain and creep recovery.

Higgins (1951) has investigated a complete analysis for inelastic columns. He used four fundamental assumptions in his analysis. Three of them are usually used in most of the structural analysis; viz, that plane sections remain plane, lateral deflections are small compared with the length, and the stress-strain relations for bending are the same as for pure tension and compression loading. The fourth and the important assumption is that the rate of inelastic strain at a given stress level depends only on the total value of inelastic strain at that time. The only disadvantage in this study might result from neglecting the creep recovery. The analysis predicted a change in the shape of the column-deflection curve with time which required an iterative method for the numerical solution. A good agreement has

been found in the comparison between Libove and Higgins studies.

Both of them in their studies have followed Shanley's approach (1952) which is based mainly on strain hardening.

An expression for the critical time in creep buckling of a column, has been given by Hoff (1954) considering a two-hinged perfect elastic idealized H-section. The elastic deformation during the buckling process was neglected compared to the creep deformation.

Two important results have been written in this study; first, the time is comparatively not largely influenced by the initial crookedness of the column. Second, the critical lifetime of the column is very sensitive to the magnitude of the applied average compressive stress. Odqvist (1954) after that generalized Hoff's formula to include the effect of primary creep beside the secondary creep. Continuation to the Hoff and Odqvist studies, Hult (1955) considered the influence of the elastic deformation on the critical time which is deduced by Hoff and Odqvist. Hult in his conclusion mentioned that the effect of elastic deformation, as well as primary creep, shorten the critical buckling time.

2-3 Methods of analysis of creep buckling

The method of analysis of creep buckling have been discussed in different ways according to the analysis assumption. For simplicity, these methods can be grouped as follows:-

a- Newtonian viscosity method: This method considers a special case of creep where the strain rate is proportional to the stress. This analysis formed the basis of several creep buckling calculations. It has the advantage of the analytical convenience but is not applied to many structural materials. Usually the initial eccentricity is represented by half a sine curve, but a sine series or a constant initial eccentricity can also be assumed. The linear relation between creep rate and stress leads to the same as the stress distribution in the elastic case. The predicted deflection increases with time but no sudden great buckling ,of the type predicted for inelastic columns, occurs. Failure may be defined in terms of a limiting stress, strain, or deflection.

b- Method based on a complete creep curve: This analysis of creep buckling is very complicated to predict the real creep curve, and some simplification is usually needed. Often the deflected shape is assumed to be sinusoidal throughout the life of the column, although it is known that this overestimates bending near the ends. This assumption greatly simplifies calculations, since only the centre cross section of the column needs then to be considered in the calculations. Such assumption could produce error, but is not likely to be serious.

c- Method depending on nonlinear stress: The first solution based on the nonlinear stress analysis was given by Marin (1947) who assumed a steady-state creep law to represent the material behaviour as follows:-

$$\dot{\varepsilon} = \beta \sigma^n \qquad (2\text{-}2)$$

where $\dot{\varepsilon}$, σ are strain rate and stress

β,n are constants may be temperature dependent.

In his study, the bending stresses have only been taken into consideration, and are not applied to cases in which the average stress is significant, for example, short column with small eccentricity. Also Johnson et al (1956) treated the column in which only bending stresses need to be considered, where the creep rate was taken to be a function of stress and time (time hardening).

2-4 Creep buckling of columns

To investigate the creep buckling for plates and shells, it is sensible to postulate the studies of creep buckling of columns. According to the Shanley's engineering hypotheses (1952), two major assumptions for a material at constant temperature can be written as follows:-

1- When a material is at some condition of stress(σ) and strain (ε) and the stress is held constant, then the strain rate ($\partial\varepsilon/\partial t$) is also constant. This rate of strain is a function of the stress and the strain only. Following to this theory, the strain change ($d\varepsilon$) occurring during an infinitesimal time interval(dt) at constant stress, may be written in the following formula;

$$d\varepsilon = f(\sigma,\varepsilon)\, dt \qquad (2\text{-}3)$$

2- When a material is at some condition of stress(σ) and strain (ε) and the stress is given an instantaneous change; then the material behaves elastically, with elastic modulus(E) being that associated with the temperature of the material. Following this theory, the strain change ($d\varepsilon$) occurring during an instantaneous stress change ($d\sigma$) can be written;

$$d\varepsilon = \left(\frac{1}{E}\right) d\sigma \qquad (2\text{-}4)$$

A general expression for strain under variable uniaxial stress can now be formulated by considering the stress-time relation to be made up of a series of infinitesimal steps. The total strain change can be written according to the last two equations as follows:-

$$d\varepsilon = f(\sigma,\varepsilon)dt + \left(\frac{1}{E}\right) d\sigma \qquad (2\text{-}5)$$

and by dividing by (dt) therefore, the average strain rate can be written as follows;

$$\left(\frac{d\varepsilon}{dt}\right) = f(\sigma,\varepsilon) + \left(\frac{1}{E}\right)\left(\frac{d\sigma}{dt}\right) . \qquad (2\text{-}6)$$

This equation constitutes a general creep law for a material under varying stress. Applications may be made for different types of columns depending on the shape and number of degrees of freedom for each case.

2-4-1 Creep buckling of a bar with one degree of freedom

Suppose two rigid bars AB and BC are jointed at B by a hinge,containing a spring subjected to creep, see Fig.(2-4). The length of each bar is(L/2), and the initial angle of rotation is (α_o) when the structure at rest,

$$\alpha=\alpha_o \ . \tag{2-7}$$

The final angle of rotation is ϕ, where;

$$\phi = 2\alpha-2\alpha_o \tag{2-8}$$

and the bending moment is,

$$M = PL/2 \ . \ \sin \alpha \ . \tag{2-9}$$

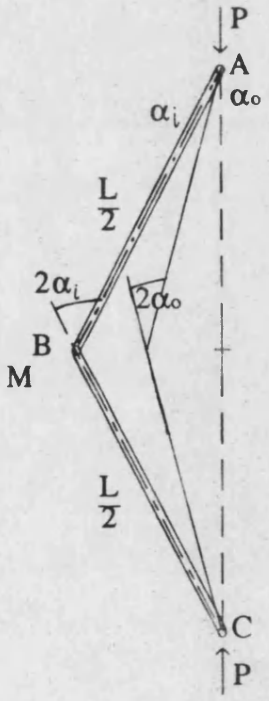


Fig.(2-4) Column with one degree of freedom

So, the constitutive equations may be written as:-

$$\phi=\phi^{(i)}+\phi^{(c)} \qquad (2\text{-}10)$$

where;

$\phi^{(i)},\phi^{(c)}$ are the instantaneous and creep deformation respectively

$$\phi^{(i)}=2(\alpha i-\alpha_o) \qquad (2\text{-}11)$$

$$\phi^{(i)}=(1-\theta).\left(\frac{M}{Me}\right)+\theta.\left(\frac{M}{Me}\right)^{k} \qquad (2\text{-}12)$$

$$\left(\frac{d\phi^{(c)}}{dt}\right)=\left(\frac{1}{\tau}\right)\left(\frac{M}{Mn}\right)^{n} \qquad (2\text{-}13)$$

where;

Me,Mn are constant bending moment depending on k,n respectively

t is the column life-time,

τ is a certain time may be chosen to be 10 s,

n,k are the temperature constants ($1\leq k\leq\cap$),

θ is a constant between 0 and 1 corresponds to a linearly elastic deflection and the pure creep law respectively.

The structure is assumed to be at rest with $\alpha=\alpha_o$. When the load, P, is applied instantaneously at t=0, the ensuing angle $\alpha=\alpha_i$ can be calculated from the following equation:-

$$2(\alpha_i-\alpha_o)=(1-\theta).\left(\frac{PL}{2M_e}\right)\sin\alpha_i+\theta.\left(\frac{PL}{2M_e}\sin\alpha_i\right)^{k} \qquad (2\text{-}14)$$

where, the initial condition $\phi^{(c)}=0$ at t=0 .

Assuming at a certain load $P=P_{cr}$, the structure becomes unstable, in sense that an infinitesimal increase of P above P_{cr} cause a finite increase in α_i. The angle corresponding to this instability load is denoted α_{cr} where α_{cr} and P_{cr} depend on the initial angle α_o.

By applying the condition $dP/d\alpha_i=0$ and if the corresponding angle is denoted (α_c) the equation (2-14) can be written as;

$$2=(1-\theta).\left(\frac{PL}{2M_e}\right)\cos\alpha_c + \theta.\left(\frac{PL}{2M_e}\right)^k k.\sin^{k-1}\alpha_c.\cos\alpha_c \qquad (2\text{-}15)$$

when $\theta=0$ $\quad \alpha_c=\alpha^*=0$

$$P_{cr}=P_E^* = \frac{4M_e}{L} \qquad (2\text{-}16)$$

then $$\frac{P}{P_E^*}=\frac{PL}{4M_e} \ . \qquad (2\text{-}17)$$

from equation (2-14) where $\alpha_i<<1$ and $\alpha_c<<1$

$$\alpha_i=\alpha_o/\left[1-(1-\theta).\left(\frac{P}{P_E^*}\right)\right] \qquad (2\text{-}18)$$

$$\alpha_c=\left[\frac{1-(1-\theta)\left(\frac{P}{P_E^*}\right)}{\theta.K\left(\frac{P}{P_E^*}\right)^k}\right]^{\frac{1}{(k-1)}} \qquad (2\text{-}19)$$

from equation (2-10) $\quad \phi^{(c)}=\phi-\phi^{(i)}$ $\qquad (2\text{-}20)$

Hence,

$$\frac{d\phi^{(c)}}{dt}=\frac{d\phi}{dt}-\frac{d\phi^{(i)}}{dt} \qquad (2\text{-}21)$$

Then,

$$\left(\frac{1}{\tau}\right).\left(\frac{PL}{2M_n}\right)^n.\sin^n\alpha =$$

$$\frac{d\alpha}{dt}\left[2-(1-\theta)\left(\frac{PL}{2M_e}\right)\cos\alpha-\theta\left(\frac{PL}{2M_e}\right)^k k.\sin^{k-1}\alpha.\cos\alpha\right] \qquad (2\text{-}22)$$

The creep buckling time t_c after which the structure becomes unstable is;

$$t_c = \tau.\left(\frac{2M_n}{PL}\right)^n$$

$$x \int_{\alpha_i}^{\alpha_c} \left[2-(1-\theta)\left(\frac{PL}{2M_e}\right)\cos\alpha-\theta\left(\frac{PL}{2M_e}\right)^k k.\sin^{k-1}\alpha.\cos\alpha\right]/\sin^n\alpha \; d\alpha$$

(2-23)

or in other formula for simplicity;

$$T.(F)^n = \int_{\alpha_i}^{\alpha_c} \left[2-(1-\theta)\left(\frac{PP}{2}\right)\cos\alpha-\theta\left(\frac{PP}{2}\right)^k k.\sin^{k-1}\alpha.\cos\alpha\right]/\sin^n\alpha \; d\alpha$$

(2-24)

where; T is equal to t_c/τ time factor dimensionless,

PP is equal to P/P_E^* load ratio dimensionless,

F is equal to PL/2Mn dimensionless load factor.

From equation (2-24) the relation between the column deformation and time can be calculated according to the applied load term (PP), the initial angle(α_o) and the material properties A,θ,n and K. Figures (2-5) & (2-6) shows the relation between time and deformation for different values of loads ratios at θ= 0.1,0.9. The initial deformation is constant for all cases. Appendix (A-1-a) includes more figures for different values of load ratio (F).

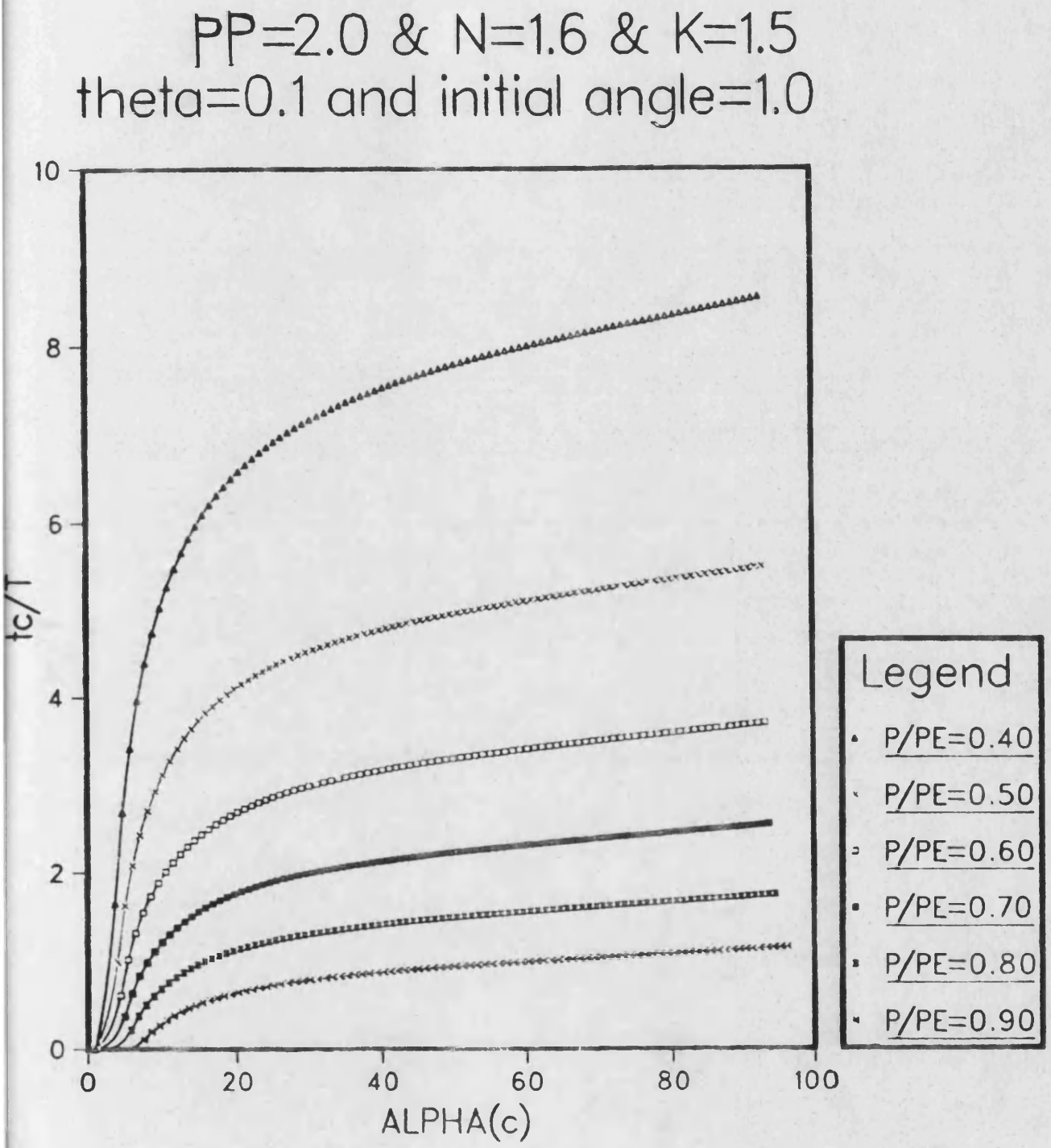


Fig.(2-5) Column with one degree of freedom

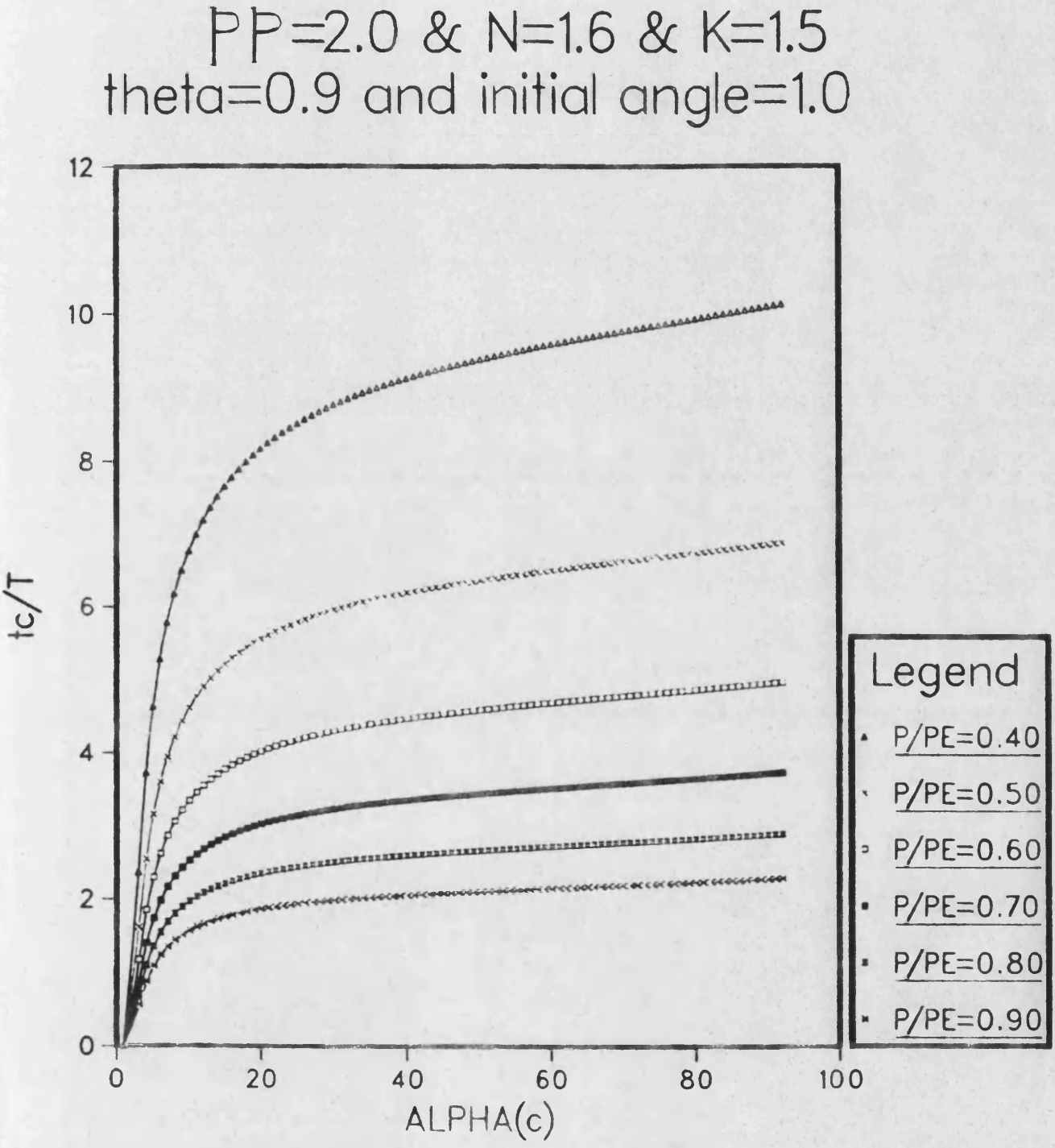


Fig.(2-6) Column with one degree of freedom

2-4-2 Two hinged H-section column

The analysis of the creep buckling for two hinged-bar is quite complex, unless the bar cross-section is of the idealized H-type as shown in Fig.(2-7). In this case, the stress distribution across the bar is statically determinate, and only its distribution along the bar remains unknown. For simplification, the deformation curve is assumed to be sinusoidal, and an approximate solution of the resulting differential equation is obtained by the collection method by Hoff (1956) , hence;

$$w_{(x,t)} = W_{(t)}.\sin\left(\frac{\pi x}{L}\right) \qquad (2\text{-}25)$$

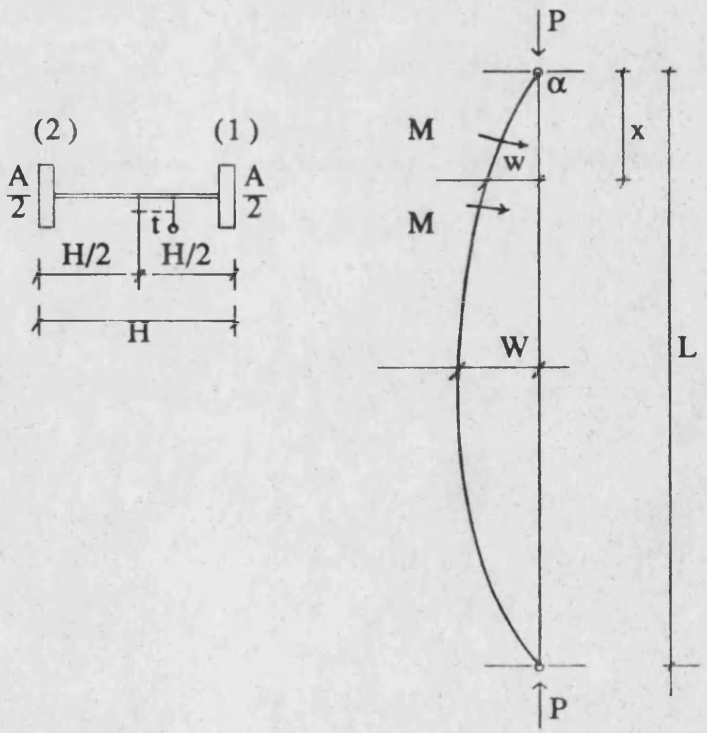


Fig.(2-7)

The equilibrium equations are,

$$\frac{\sigma_1 A}{2} + \frac{\sigma_2 A}{2} = P \tag{2-26}$$

$$\frac{\sigma_1 AH}{4} - \frac{\sigma_2 AH}{4} = Pw. \tag{2-27}$$

With $\sigma_m = P/A$ $\delta = 2w/H$

$\Delta = 2W/H$ $\zeta = \pi x/L$

there results $\sigma_1 = \sigma_m.(1+\Delta \sin\zeta)$ for inner flange

$\sigma_2 = \sigma_m.(1-\Delta \sin\zeta)$ for outer flange (2-28)

and the compressive strain of the two flanges are;

$$\varepsilon_1 = \varepsilon_o - \frac{1}{2}H\left(\frac{\partial^2 w}{\partial x^2} - \frac{\partial^2 w_o}{\partial x^2}\right) \tag{2-29}$$

$$\varepsilon_2 = \varepsilon_o - \frac{1}{2}H\left(\frac{\partial^2 w}{\partial x^2} - \frac{\partial^2 w_o}{\partial x^2}\right) \tag{2-30}$$

$$\varepsilon_1 - \varepsilon_2 = 2(\pi H/L)^2(\Delta - \Delta_o)\sin\zeta \tag{2-31}$$

where ε_o is the compressive strain of the centre line and $w_o(x)$ is the initial deflection. Therefore, according to the constitutive creep equations where in the inner flange the stress, σ_1, is increasing and in the outer flange the stress, σ_2, is decreasing during the creep deformation then;

$$\frac{d\varepsilon_1}{dt} = \frac{1}{E}\frac{d\sigma_1}{dt} + \frac{d}{dt}\left(\frac{\sigma_1}{\sigma_k}\right)^k + \frac{1}{\tau}\left(\frac{\sigma_1}{\sigma_n}\right)^n \tag{2-32}$$

$$\frac{d\varepsilon_2}{dt} = \frac{1}{E}\frac{d\sigma_2}{dt} + \frac{1}{\tau}\left(\frac{\sigma_2}{\sigma_n}\right)^n \operatorname{sgn}\ \sigma_2 \tag{2-33}$$

where the signum function is defined by

$$\begin{aligned} \text{sgn } x &= -1 \qquad x < 0 \\ &\quad\ 0 \qquad x = 0 \\ &\quad +1 \qquad x > 0 \end{aligned}$$

The differential equation can be written as follows;

$$\frac{d\Delta}{dt}\left[1-\frac{\sigma_m}{\sigma_E}-\left(\frac{K.E}{2\sigma_E}\right)\left(\frac{\sigma_m}{\sigma_k}\right)^k.(1+\Delta)^{k-1}\right]=\pi\left[(1+\Delta)^n-|1-\Delta|^n \text{sgn}(1-\Delta)\right] \qquad (2\text{-}34)$$

The creep buckling will occur after a certain finite time, and (tc) which can be obtained by the following integration;.

$$t_c=\frac{1}{\Pi}\int_{\Delta_o}^{\Delta_c}\frac{\left[1-\frac{\sigma_m}{\sigma_E}-\frac{K}{2}\frac{E}{\sigma_E}\left(\frac{\sigma_m}{\sigma_k}\right)^k.(1+\Delta)^{k-1}\right]}{\left[(1+\Delta)^n-|1-\Delta|^n \text{sgn}(1-\Delta)\right]}d\Delta \qquad (2\text{-}35)$$

and

$$\Pi=\frac{1}{2}\cdot\frac{E}{\sigma_E}\left(\frac{\sigma_m}{\sigma_n}\right)^n\frac{1}{\tau}$$

$$\sigma_E=(\pi H/2L)^2.E$$

$$\Delta(o)=\Delta o/\left[1-\frac{\sigma_m}{\sigma_E}\right] \qquad (2\text{-}36)$$

Figures (2-8) and (2-9) show the relation between time factor (t_c/τ) and the angle of rotation (α_i) for different values of stress ratios (σ_m/σ_k) at initial angle $\alpha_i = 1^\circ$

More results are given in Appendix(A-1-b) for different values of (σ_m/σ_k).

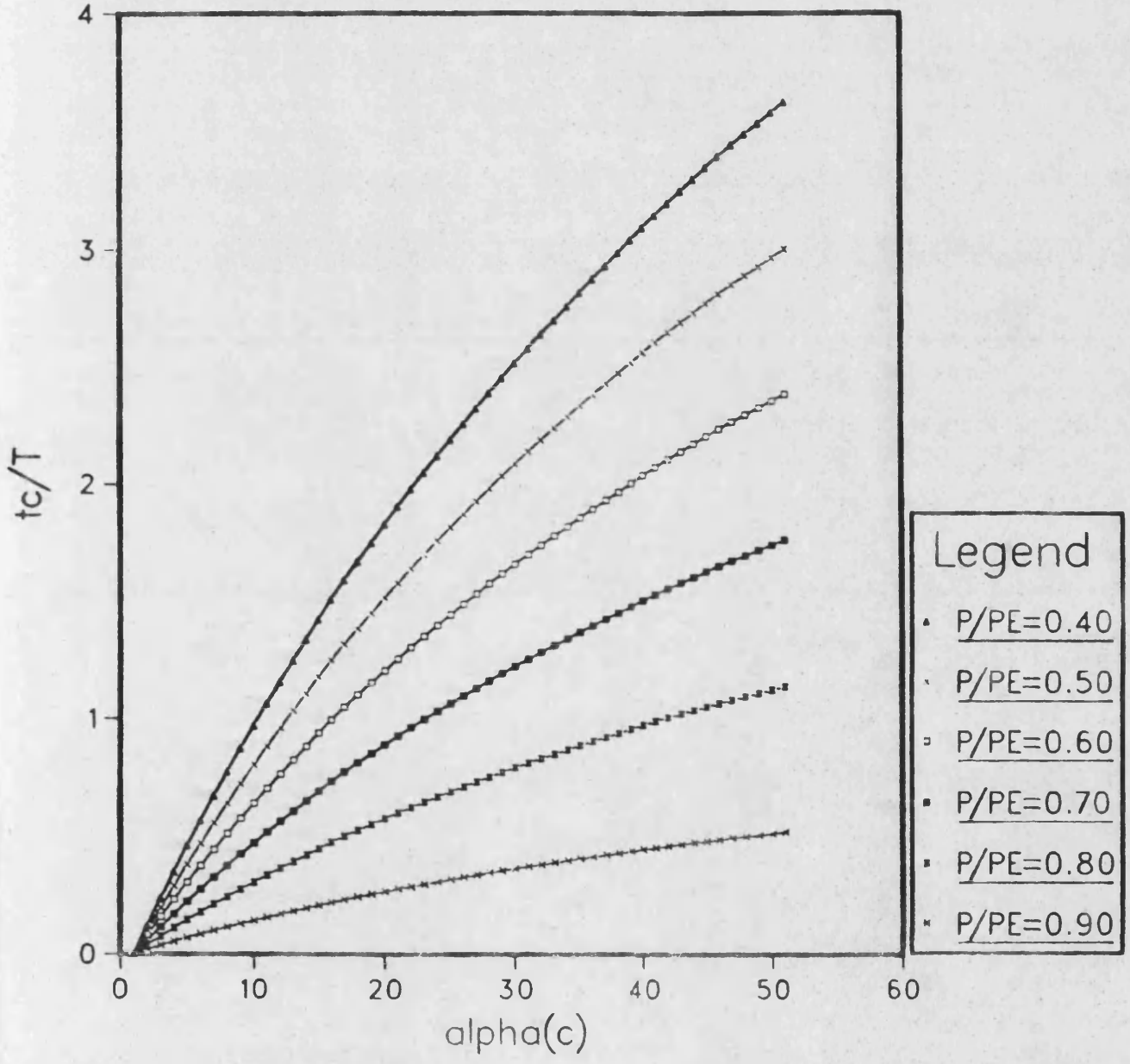


Fig.(2-8) Two hinged H-section column

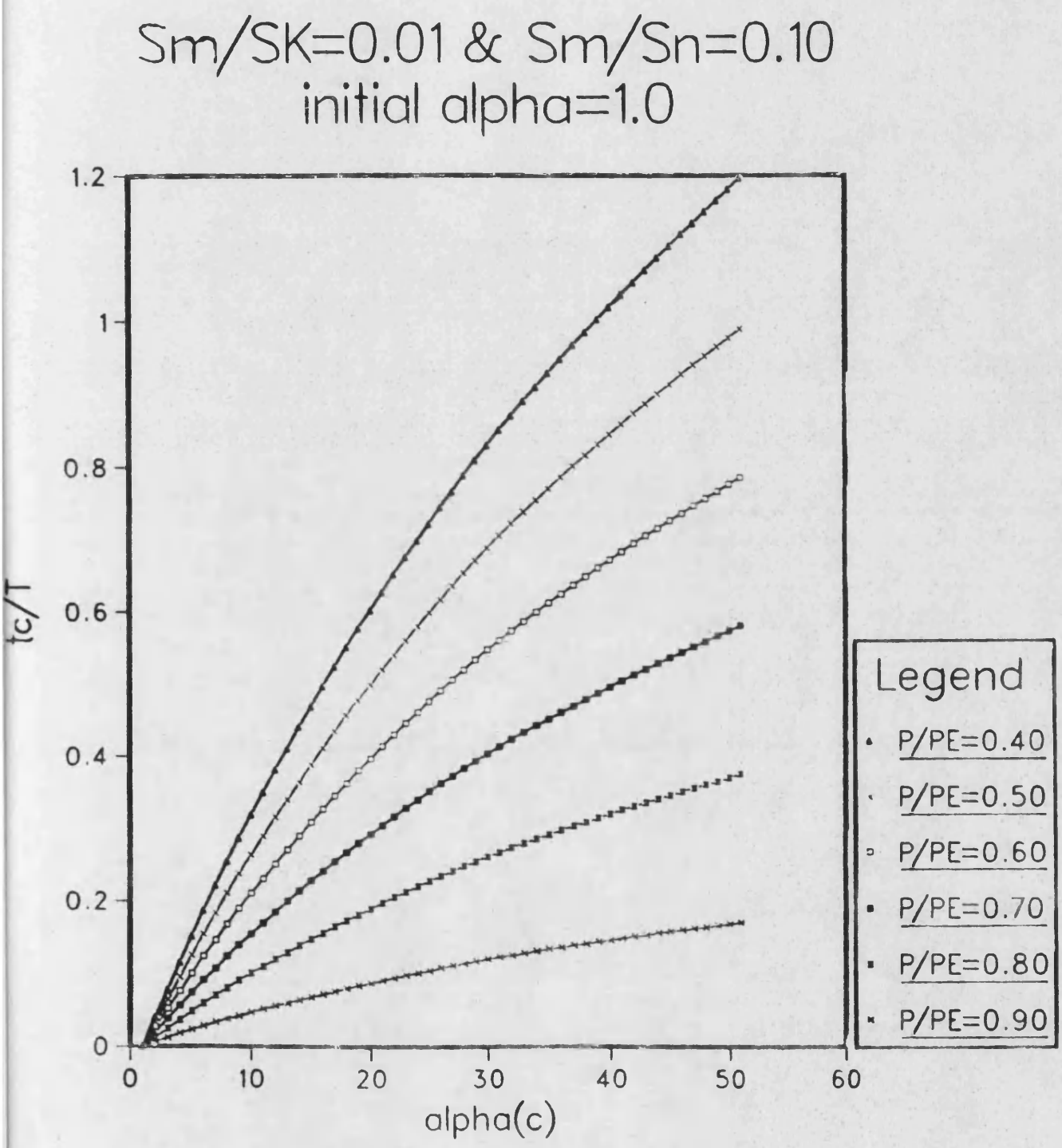


Fig.(2-9) Two hinged H-section column

2-4-3 Unstable case

As shown in Fig.(2-10-a), a spring(λ) has been attached to the bar(L) to keep it vertical. Small imperfection(Δ_o) has assumed at time t=0 .

The tension force in the spring can be easily written as;

$$F=\lambda(\Delta-\Delta_0) \qquad (2\text{-}37)$$

or $$F=\lambda.L(\sin\alpha-\sin\alpha_o) \qquad (2\text{-}38)$$

Comparing the spring model with bar that has one hinge as shown in Fig.(2-10-b) then the relation between the initial angle α_o and the angle of rotation ϕ, for both cases, can be written as follows.

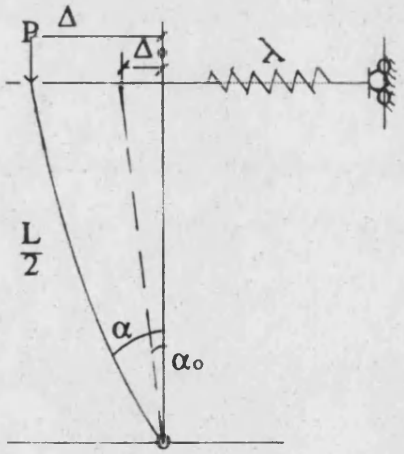


Fig.(2-10-a)

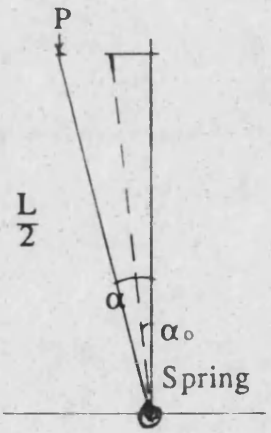


Fig.(2-10-b)

$\phi = \phi^{(i)} + \phi^{(c)}$

$\phi = 2(\sin\alpha - \sin\alpha_o)$

$M = (FL/2)$

$M = (PL/2).\tan\alpha$

$\phi = \phi^{(i)} + \phi^{(c)}$

$\phi = 2(\alpha - \alpha_o)$

$M = (PL/2).\sin\alpha$

Using equations (2-12) and (2-13) of the creep deformation, therefore;

$$2(\sin\alpha_i-\sin\alpha_o)=(1-\theta)\left(\frac{PL}{2M_e}\right)\tan\alpha+\theta\left(\frac{PL}{2M_e}\right)^k\tan^k\alpha \qquad (2\text{-}39)$$

When $dP/d\alpha_i=0$ and α_c is the corresponding angle, hence;

$$2\cos\alpha=(1-\theta)\left(\frac{PL}{2M_e}\right)\sec^2\alpha+\theta.K.\left(\frac{PL}{2M_e}\right)^k\tan^{k-1}\alpha.\sec^2\alpha \qquad (2\text{-}40)$$

Substituting equation (2-13) and (2-40) in equation (2-21) therefore,

$$\frac{d\alpha}{dt}\left[2\cos\alpha-(1-\theta)\left(\frac{PL}{2M_e}\right)\sec^2\alpha-\theta.K.\left(\frac{PL}{2M_e}\right)^k\tan^{k-1}\alpha.\sec^2\alpha\right]$$

$$=\frac{1}{\tau}\left(\frac{PL}{2M_e}\right)^n\tan^n\alpha \qquad (2\text{-}41)$$

The creep buckling time t_c can be calculated as;

$$\frac{t_c}{\tau}=\left(\frac{2M_n}{PL}\right)^n$$

$$x\int_{\alpha_i}^{\alpha_c}\frac{2\cos\alpha-(1-\theta).\left(\frac{PL}{2M_e}\right)\sec^2\alpha-\theta.K.\left(\frac{PL}{2M_e}\right)^k\tan^{k-1}\alpha.\sec^2\alpha}{\tan^n\alpha} \qquad (2\text{-}42)$$

A relation between the column's lifetime and the deflection is shown in Figures(2-11) and (2-12). Appendix (A-1-c) includes more figures for different values of (PL/2Mn) and (θ). Also a list of the computer programs which have been used in this calculation is shown in Appendix A-2

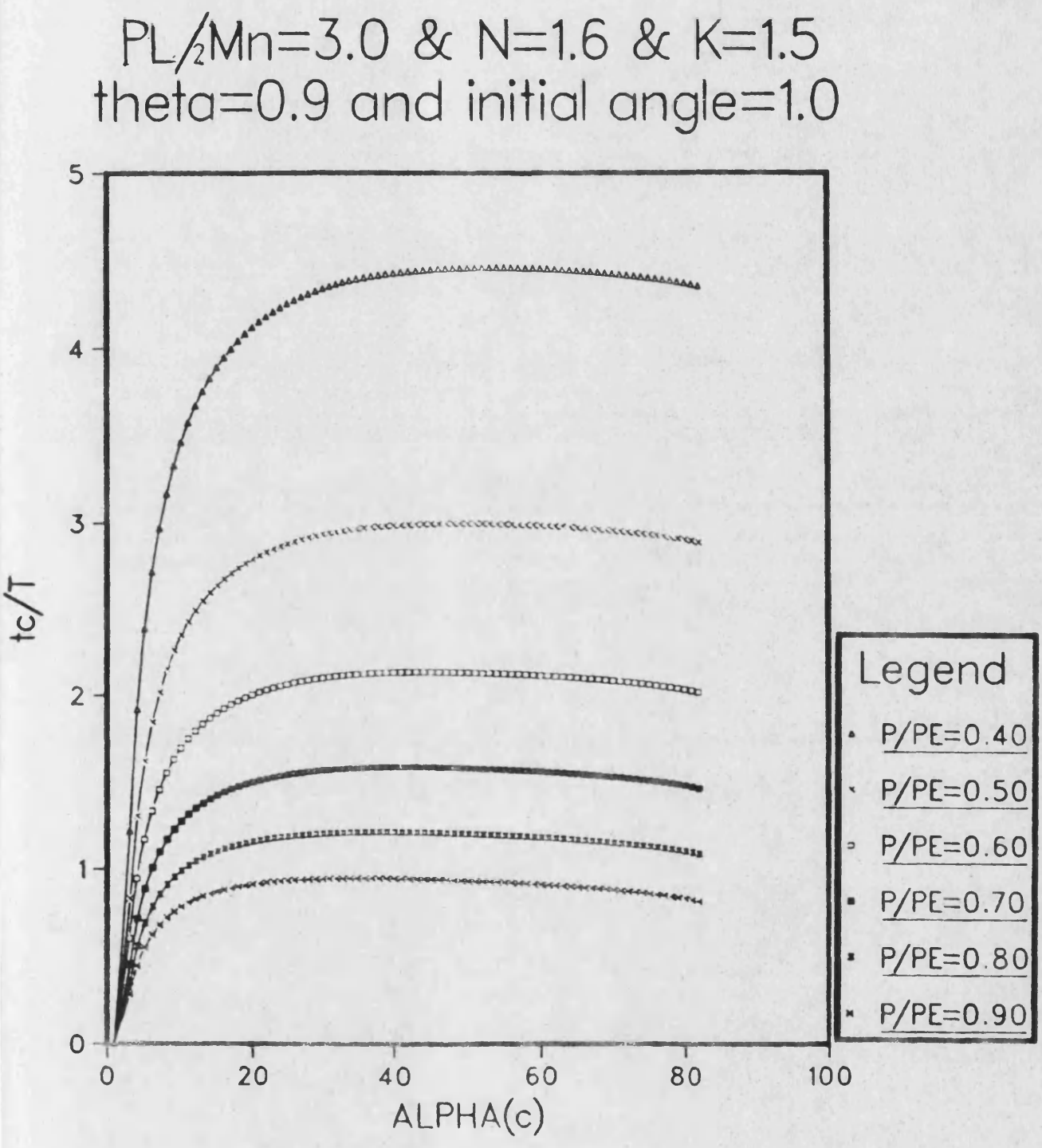


Fig.(2-11) Unstable case

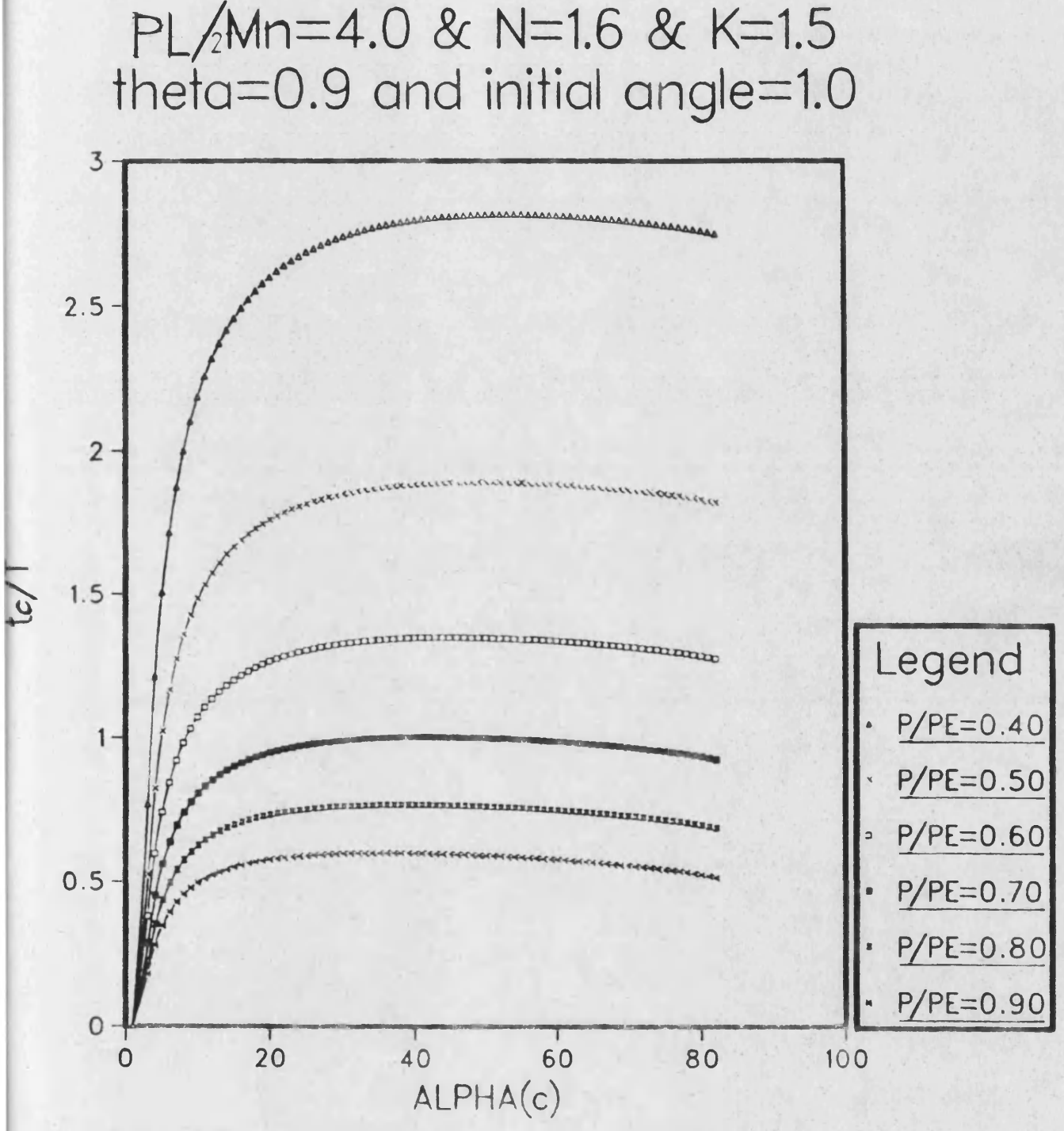


Fig.(2-12) Unstable case

2-5 Creep buckling of plates and shells

The studying of creep buckling for plates and shells is more complex than in case of pin-ended columns. A similar approach relating creep buckling to the existing inelastic buckling solutions for the pin-ended column is indicated for the plates and shells. The inelastic buckling solutions can be obtained in turn by replacing the elastic modulus, in the elastic buckling equations, by an effective modulus of the plates.

The effective modulus involves both the tangent modulus and the secant modulus of the plates is shown in Fig.(2-13). The effective modulus equals the tangent modulus when the longitudinal bending predominates the deformation. On the other hand, when longitudinal bending is small compared with other types of distortion, then the secant modulus is the effective modulus.

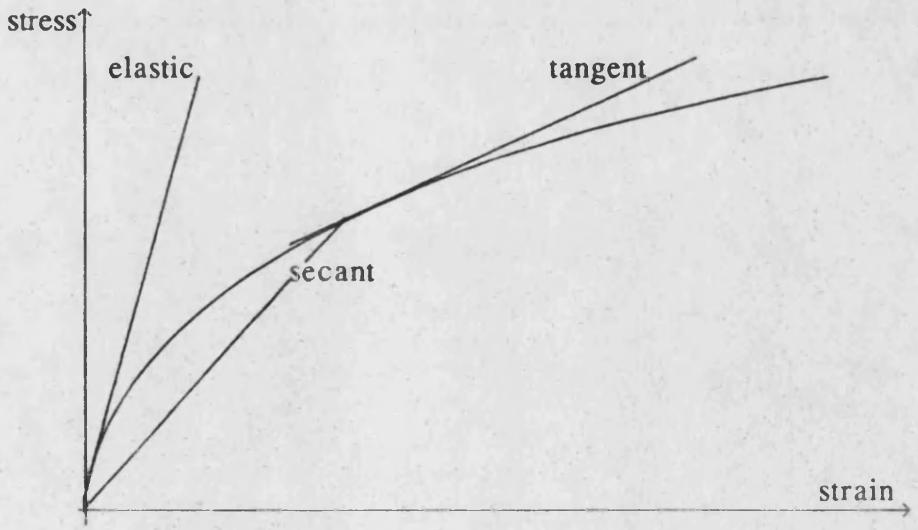


Fig.(2-13)

2-6 Creep buckling of grid shells

In section 1-6 it was stated that grid shells are system of interconnected arches and therefore grid shells exhibit structural behaviour part way between that of arches and that of shells. This also applies to their deflection due to creep. The creep behaviour of an individual grid shell member will be similar to that of a member can only be calculated if one knows the deformation of the structure as a whole. The structure is statically indeterminate where as a single pin ended column is statically determinate. In order to predict the creep behaviour of grid shells it is necessary to:

a) Assume a relationship between bending moment, curvature and time.

b) Employ an analysis technique which steps through time. Equilibrium of forces and moments has to be maintained and the increment of displacements must be compatible with the increments of creep and elastic strains.

2-7 Conclusion

General conclusions can be written for creep buckling of columns as follows;

a) Column with one degree of freedom.

- The lifetime of this column depends on four different factors as follows;
  1) α the initial angle,
  2) n,k temperature functions,

3) P,F load ratios,

4) θ deflection constant.

- The load ratios have a significant effect on the column lifetime
- The temperature and deflection functions have less influence on the relation between the deformation and time.
- The initial angle of rotation up to 5° does not make a big difference.

b) Two hinged H-section columns.

The following factors have been found effective in shortening the lifetime of such column against the deformation.

1) Increasing the loads ratios (P/PE) and ($\sigma m/\sigma n$), where the(PE) is the Euler load.
2) Decreasing the temperature functions n,k
3) Finally, the initial deformation does not have a noticeable influence on the relation between the time and the deformation.

c) Unstable structure.

The critical time of the column can be predicted more easily than the other two cases as shown in Figures (2-12) and (2-13).

CHAPTER THREE

FUNDAMENTAL PRINCIPLES

Certain principle theories will be used in this thesis, such as the theory of minimization of the total potential energy and the principle of virtual work as well as the finite element techniques and the spline function. Therefore, it is important to discuss briefly these principles.

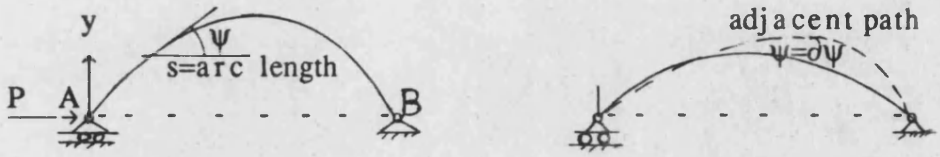
3-1 The calculus of variations

The calculus of variations was conceived by Bernoulli, Euler, and Lagrange for the solution of a certain group of problems in differential geometry and physics. This branch of mathematics together with the theory of linear differential equations form the backbone of the mathematical treatment of various problems in statics and dynamics. Knowledge of the fundamental features of the calculus of variations is indispensable for a thorough understanding of the nature of the energy method. The most important advantage of this advanced mathematical analysis, is dealing with structural elements with different boundary conditions and this is the case in the present study.

The calculus of variations enables problems posed in terms of minimization of certain quantities to be reformulated as differential equation.

For instance, the differential equation of a plane curve, can be provided by using the calculus of variations as follows;

Assuming a plane curve with length=S and its bending stiffness=β as shown below;



The energy equation of this curve is;

$$I = \frac{\beta}{2}\int\left(\frac{d\psi}{ds}\right)^2 ds + P\int\cos\psi\, ds \qquad \text{is minimum} \qquad (3\text{-}1)$$

After a small displacement $d\psi$, the energy equation will take the following form;

$$I+\partial I = \frac{\beta}{2}\int\left(\frac{d(\psi+d\psi)}{ds}\right)^2 ds + P\int\cos(\psi+d\psi)\, ds \qquad (3\text{-}2)$$

and with some simplification the change in the energy can be calculated;

$$\partial I = \frac{\beta}{2}\int 2\,\frac{d\psi}{ds}\,\frac{d(d\psi)}{ds}\, ds - P\int\partial\psi\,\sin\psi\, ds. \qquad (3\text{-}3)$$

If I is a minimum then $\partial I=0$ to the first order in $\partial\psi$ and by using

integration by parts then the ∂I must $=0$ for any $\partial\psi$;

$$\text{i.e.} \qquad \beta \frac{d^2\psi}{ds^2} + P \sin \psi = 0 \qquad (3\text{-}4)$$

So, this is the plane curve differential equation.

The calculus of variations is described in details in Forsyth (1960), and for more applications on two-dimensions and surface problems under different types of boundary conditions can be seen in Williams (1987).

3-2 Principle of virtual work

The principle of virtual work is related to two distinct and separate systems in which the first is a set of forces in equilibrium, with the external forces (P) and internal stresses (σ), and the second is a set of geometrically compatible deformations, with the displacements (Δ) and strains (ε). The principle states that for any system in equilibrium, the external virtual work must be equal to the internal virtual work, hence;

$$\Sigma P.\Delta = \int_v \sigma.\varepsilon \, dv \qquad (3\text{-}1)$$

Where; P and σ represents the equilibrium system

Δ and ε represents the compatible deformations at a certain deformed configuration

From the above relation, one of the systems always relates to

a real or actual structure in which some sort of solution is required, while the other is an imaginary or virtual system. Therefore, it is possible to have the option of establishing;

a) Theorem of virtual forces in which a real system of displacements and strain is coupled to a virtual system of forces and stresses applying equation (3-1), then

$$\Sigma(\text{Virtual external forces}).(\text{Actual displacements}) = \int_v (\text{Virtual stresses}).(\text{Actual strains})\, dv \qquad (3\text{-}2)$$

b) Theorem of virtual displacements in which a real system of forces and stresses is coupled to a virtual system of displacements and strains. Again, using equation (3-1),

$$\Sigma(\text{Actual external forces}).(\text{Virtual displacements}) = \int_v (\text{Actual stresses}).(\text{Virtual strains})\, dv \qquad (3\text{-}3)$$

Equation (3-2) is used to calculate displacements and leads to the unit load theorem while equation (3-3) is used to calculate external forces and leads to the unit displacement theorem.

Therefore the virtual work analysis is considered more general in structural problem applications especially with the nonlinear problems such as shells as it will be shown later.

It is important to notice that the virtual work theorem makes no assumptions regarding material behaviour. Thus the theorem can

be applied to problems in elasticity, plasticity and problems involving creep.

3-3 Principle of minimum total potential energy

The total potential energy of a system is defined as,

$$\phi = U + W \tag{3-4}$$

in which W is the potential energy of the external force, P, in the deformed configuration and is defined as.

$$W = -\Sigma\ P.\Delta \tag{3-5}$$

and U is the strain energy of the deformed structure and is given by,

$$U = \int_v \left(\int \sigma\ d\varepsilon \right) dv \tag{3-6}$$

If the deformed system is now taken as the real or the actual system and a set of small geometrically compatible displacements $\partial\Delta$, as the virtual system, then by virtue of equation (3-3) it is possible to establish that,

$$\Sigma\ P.\partial\Delta = \int_v (\sigma.\partial\varepsilon)\ dv\ . \tag{3-7}$$

However, due to the imposed virtual displacements, the actual system will also undergo a change of a total potential energy of

$$\partial\phi = \partial U + \partial W$$

then $$\partial\phi = \int (\sigma.\partial\varepsilon)dv - \Sigma P.\partial\Delta \qquad (3\text{-}8)$$

By comparing equation (3-7) with equation (3-8), it can be concluded that

$$\partial\phi = 0 \qquad (3\text{-}9)$$

in other words, the total potential energy of a system in equilibrium is stationary, and also because it can be further proved that the energy is always a minimum for stable structures, it is called the principle of the minimum potential energy. The principle can also be alternatively expressed as "of all compatible displacements satisfying given boundary conditions, those which satisfy the equilibrium conditions make the total potential energy an assumed a stationary value". A solution satisfying both equilibrium and compatibility is of course the correct solution for a linear elastic problem.

However, such solutions are difficult, if not impossible, to work out for the majority of cases and researchers have to resort approximate solutions by assuming compatible displacements with undetermined parameters such that the total potential energy of the system is a minimum.

It follows that if a set of trial displacement functions with unknown parameters Δ_i is used to approximate the actual displacements of a system, then it is possible to determine such Δ_i by minimizing the total potential energy, i.e. by performing the operation;

$$\frac{\partial \phi}{\partial \Delta_i} = 0 \qquad (3\text{-}10)$$

this relation has been used extensively in the derivation of finite element stiffness matrices as it will be explained in the following section.

3-4 Numeral analysis approach

The solution of the differential equations of a complex structure such as shells, might only be possible by the numerical methods. The finite differences and the finite element techniques are considered the most powerful tools for the structural engineering problems.

3-4-1 Finite difference analysis

All the methods of structural analysis are essentially concerned with solving the basic differential equations of equilibrium and compatibility supplemented by equations describing material behaviour. Analytical solutions are limited to the cases when the load distribution, section properties and boundary conditions can be described by mathematical expressions. But for complex structures numerical methods are in general a more practical means of analysis.

The numerical solution by finite differences generally requires replacing the derivatives of a differential equation by difference expressions of the function at the nodes.

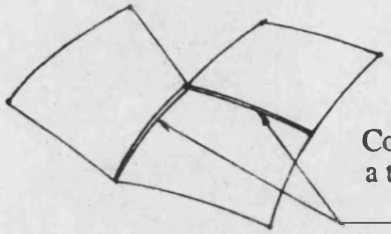
The numerical solution by finite differences governing the displacement (or stress) is applied in a difference form at each node, relating the displacement at the given node and nodes in its vicinity to the external applied load. This usually provides a sufficient number of simultaneous equations for the displacements (or stresses) to be determined. The finite difference coefficients of the equations applied at the nodes on ,or close to, the boundary have to be modified compared with the coefficients used at interior points in order to satisfy the boundary conditions of the problem.

3-4-2 Finite element analysis

The main disadvantage of the finite difference is the difficulty encountered in providing extra nodes in areas where greater accuracy is required. Also, this analysis does not ,of itself, provide a method of interpolation between nodes. On the other hand the finite element method starts with the question of interpolation. The finite element method has been applied widely to many structural and non-structural problems, see Zienkiewicz (1967),(1971).

In the finite element method, the actual continuum is replaced by an equivalent idealized structure composed of discrete elements refereed to as finite elements. Each discrete element is required to deform reasonably and similarly to the deformation developed in the corresponding region of the structure.

Continuity in displacement and slope between each two neighboring nodes should be adjusted, in some problems the curvature also is needed to be continuous.



Continuity must be granted at every neighbouring edges of discrete element

The main aim of this technique is to converge to the true solution as the number of elements is increased. This aim can be achieved by choosing a suitable displacement function. Many authors such as Coates et al (1972) and Ghali et al (1978) in their books mentioned that a good displacement function should satisfy the following conditions;

a) The displacement function and its derivatives should be continuous within the element.
b) Allow nodal displacements caused by rigid body translations and rotations to occur without changing the strain energy within the element.
c) Allow the strain within the element to be uniform for all states.
d) Maintain the internal compatibility within the element and also maintain compatibility and displacements between adjacent elements at the nodes and along the boundaries.

A stiffness matrix is written to relate the nodal forces to the nodal displacement parameters, using, as mentioned before, the principle of virtual work or the minimum of total potential energy.

A set of simultaneous algebraic equations is formed, the solution of which gives the nodal displacements, which in turn are used to determine all the internal stresses.

In the application of the finite element on curves and surfaces, the stiffness matrix of the element will deal with the coordinates of the control points that cover the element or the patch. The overall stiffness matrix will be formed from the overlapping between the control points which will cover all the surface. Therefore an interpolation method must be needed to guarantee the smoothness of the curve or the surface.

CHAPTER FOUR

GEOMETRIC FUNDAMENTALS

4-1 The spline function as an interpolation method

The finite element method requires displacement functions which allow interpolation of displacements between nodes within an element. In this section the use of the spline function for interpolation will be discussed.

The modern mathematical theory of spline approximation was introduced by Schoenberg (1946). In that paper he developed splines for use in a new approach to statistical data smoothing.

The application of splines to surface interpolation uses methods which were first developed for plane curves. Therefore the application of splines to plane curves will be discussed prior to consider surfaces. Complex curved surfaces such as car bodies or aerodynamic surfaces of aircraft may need a definite mathematical representations. As described by Faux and Pratt (1979) , Riesenfeld (1973) and Gordon (1979) , the procedure used for the mathematical representation of surfaces such as these, commonly runs roughly as follows;

a) Two notional sets of lines are envisaged to lie in the surface, one set running fore and aft, the other transversely. This

network of lines defines a number of 'topologically rectangular' patches, each of which, for a smooth surface will be bounded by four smooth continuous curves.

b) The coordinates of the intersections of this notional mesh are measured from a model or a set of cross-sectional drawings of the surface.

c) An interpolatory technique is used to establish firm mathematical specifications of the two sets of lines comprising the network.

d) Each mesh cell of the network now has four well-dimensional boundaries, and the interior of the cell is filled by using two-dimensional interpolation.

The mathematical details of the process described are simplest when the curved surface is defined with respect to a flat plane, the (x,y)-plane, the mesh being composed of lines of constant x and of constant y. The fitting of plane curves can be achieved by applying one of the classical methods of numerical analysis such as Lagrange's method, Hermite's method and the Least squares polynomial. Never the less, each of those methods has one at least of the following major disadvantages;

1- Oscillatory tendencies.

2- Inability to produce a smooth curve.

3- A requirement for gradient values at the data points.

4- Discontinuity in slope (the second derivative).

By using a low-degree polynomial the problem of oscillation

can be reduced and using a cubic function, the continuity of the first and second derivative can be provided.

4-1-1 Cubic spline

The generation of a cubic spline $\phi(x)$ which fits the data points (x_i,y_i), i=1,2, ..,n can be written as follows;

a) $\phi(x)$ is a polynomial of degree$\leq$3 in each interval $(x_{i-1} \leq x \leq x_i)$

b) $\phi(x_i) = y_i$

c) $\phi'(x)$ and $\phi''(x)$ are continuous at the nodes $x_1,x_2,..,x_{n-1}$

The practical implementation of cubic splines may be started by considering initially only a single span of the spline, whose width is $h_i=x_i-x_{i-1}$, as shown in Fig.(4-1). Since $\phi(x)$ is cubic. $\phi'(x)$ is quadratic and $\phi''(x)$ is linear in x over this span. If $\phi''(x_{i-1})=S_{i-1}$ and $\phi''(x_i)$ are given at any point on the span by the linear interpolation formula;

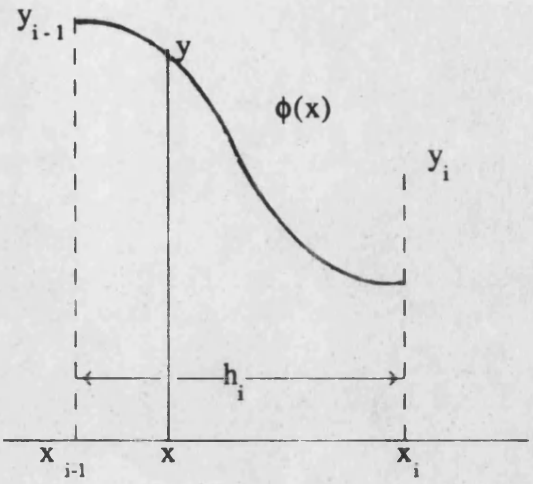
$$\phi''(x)=\frac{S_{i-1}(x_i-x)+S_i(x-x_{i-1})}{h_i} \qquad (4\text{-}1)$$

where,

$\phi''(x)$ is straight line and by integrating twice this will result in,

$$\phi(x)=\frac{S_{i-1}(x_i-x)^3+S_i(x-x_{i-1})^3}{6h_i} + C_1x + C_2 \qquad (4\text{-}2)$$

where the constants of integration C_1 and C_2 may be evaluated from the end conditions $\phi(x_{i-1})=y_{i-1}$, $\phi(x_i)=y_i$ to give,



Fig(4-1)

$$\phi(x)= \frac{S_{i-1}(x_i-x)^3}{6h_i} + \frac{S_i(x-x_{i-1})^3}{6h_i}$$

$$+\left\{\frac{y_{i-1}}{h_i} - \frac{S_{i-1}.h_i}{6}\right\}(x_i-x)+\left\{\frac{y_i}{h_i} - \frac{S_i.h_i}{6}\right\}(x-x_{i-1}) \qquad (4\text{-}3)$$

This expresses the interpolating cubic over the span $x_{i-1} \le x \le x_i$ in terms of two known: y_{i-1},y_i, and two unknown: S_{i-1},S_i.

The values of S_{i-1} and S_i can be determined by using the property of first derivative continuity at the nodes of the spline. On differentiating equ.(4-3) with respect to x and setting $x=x_i$ and after some simplification then the equation will lead to;

$$\phi'(x_i)=\frac{y_i-y_{i-1}}{h_i} + \frac{S_i.h_i}{3} + \frac{S_{i-1}.h_i}{6} \qquad (4\text{-}4)$$

Now by replacing (i) by i+1 throughout (4-3) the cubic which

interpolates the next span, ($x_i \leq x \leq x_{i+1}$) can be obtained. By differentiating this and setting $x=x_i$ then,

$$\phi'(x_i)=\frac{y_{i+1}-y_i}{h_{i+1}} - \frac{S_i.h_{i+1}}{3} - \frac{S_{i+1}.h_{i+1}}{6} \qquad (4\text{-}5)$$

where $h_{i+1}=x_{i+1} - x_i$

Since $\phi'(x)$ is to be continuous across $x=x_i$ the right hand sides of (4-4) and (4-5) must be equal. this gives,

$$h_iS_{i-1}+ 2(h_i+h_{i+1})S_i+ h_{i+1}S_{i+1}=\frac{6(y_{i+1}-y_i)}{h_{i+1}} - \frac{6(y_i-y_{i-1})}{h_i} \qquad (4\text{-}6)$$

which is linear in the three unknowns S_{i-1}, S_i and S_{i+1}. An equation of this type for each of the (n-1) internal nods $x_1,x_2,$ $,x_{n-1}$ can also be written. But there are (n+1) values of S to be calculated, namely $S_0,S_1,.....,S_n$, and therefore two additional relations must be needed to complete the system and to compute the spline unambiguously.

By assuming that each end span is a quadratic function, and has a constant second derivative, the two extra constraint equations are $S_0=S_1$ and $S_{n-1}=S_n$. This assumption is the only one fairly obvious method to solve the function without specifying any derivative values. A more satisfactory way of doing this, however, is to set up a single cubic over the double span (x_0«x« x_2) which is also required to interpolate the point (x_1,y_1), and similarly at the other end of the spline. The resulting equations are,

$$h_2 S_0 - (h_1+h_2)S_1 + h_1 S_2 = 0$$

$$h_n S_{n-2} - (h_{n-1} + h_n)S_{n-1} + h_{n-1} S_n = 0 \qquad (4\text{-}7)$$

It should by now be apparent that splines are smooth yet flexible and have many applications in curve fitting and design. They have great advantages where second order continuity is attained (so that curvature is continuous) and that a little priori derivative information is required in their construction. They do have limitations, however, among which are;

a) A local modification involves the recomputation of the entire splines,

b) A spline as treated here will not cope with a vertical tangent,

c) Oscillation problems may arise in the approximation of a curve with a discontinuity in its second derivative, for example the continuation of a straight line by a circular arc.

Of these three problems, the first can be avoided by the use of a B-splines, as will be explained in the section 4-1-2, while the second can be overcome and the third be alleviated by the use of parametric splines.

4-1-2 B-spline

The last section shows how a cubic spline can be constructed over (n) spans. By virtue of the continuity conditions imposed, the only data requirements are the function values y_i to be interpolated at the (n+1) nodes, together with just two extra

items of information, making (n+3) items of data in all.

Bearing this in mind, the construction of a cubic spline $\phi(x)$ is considered to have the properties $\phi(x)=\phi'(x)=\phi''(x)=0$ at each end. These six items of data should enable the unique spline over three spans to be computed . This is indeed so, but the spline proves to be $\phi(x)=0$, which do not neglect to fit the end conditions and has the required continuity at the nodes. The case of four spans is much more fruitful, then an additional item of data is needed, and can therefore specify a non-zero function value at an internal nodes, which ensure that the spline is not identically zero. The general form of the resulting curve is as shown in Fig.(4-2), where the spline has been extended from its end-points x_{i-4}, x_i by straight lines along the X-axis.

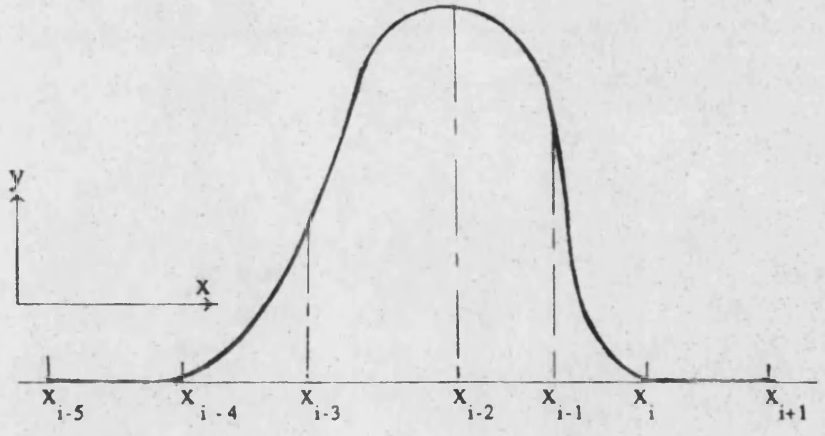


Fig.(4-2)

This analysis can also be understood from the draughtman's

spline as shown in Fig. (4-3). Where every part of it is assumed to be a piece of elastica. Drawing the deflection line which is considered as a small deflection problem, the shear force equation can be written in the form;

$$\frac{dM}{dx} = EI \frac{d^3y}{dx^3} \qquad (4\text{-}8)$$

then;

$$y = \frac{P}{EI} \frac{x^3}{6} + Ax^2 + Bx + C \qquad (4\text{-}9)$$

where; A,B and C are constant, and by satisfying the end conditions an extra item of data (like the end forces) is needed to get the required shape of the B-spline.

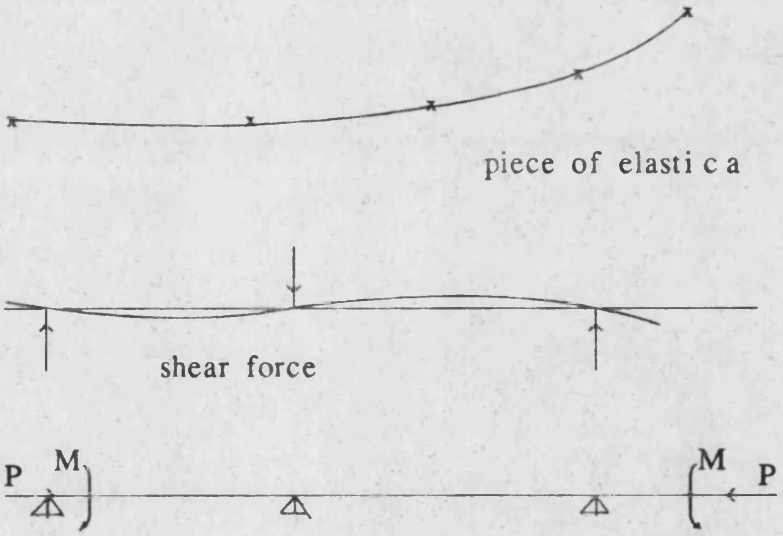


Fig. (4-3) draughtman's spline

The result is a cubic spline over an indefinite number of spans, but which departs from zero only over precisely four of those spans. Such a function is called a B-Spline (or fundamental spline) of order 4 (or degree 3). A B-spline is said to be a spline of minimal support, its support being the number of spans over which a spline is non-zero.

The practical importance of B-splines is due to the property that any spline of order m on a set of nodes $x_0,x_1,....,x_n$ can be expressed as a sum of multiples of B-splines defined on the same node set extended by (m-1) additional nodes at each end of the range, which may be chosen arbitrarily; $x_{-m+1},x_{-m+2},...,x_{-1}$ and $x_{n+1},...,x_{n+m-1}$. It is possible to construct m+n-1 successive B-splines on the extended node set, each of which is non-zero over just m consecutive spans hence;

$$\phi(x)=\sum_{i=1}^{m+n-1} c_i \, M_{mi}(x) \qquad (4\text{-}10)$$

where, $\phi(x)$ is any spline of degree (m-1) on the original node set ,$M_{mi}(x)$ is the spline on the extended node set which is non-zero for $(x_{i-m}<x<x_i)$ and c_i are numerical coefficients.

The result is very significant because the B-spline is only locally non-zero. If a spline of degree m-1 is expressed in terms of B-splines, therefore changing the coefficient of one of the B-splines will alter precisely m spans of the curve, without affecting its continuity properties. This means it is possible to

make any local modifications to the curve without having to recompute it completely.

4-2 Parametric cubic curves

4-2-1 The cubic B-spline curve segment

Using the spline function as an interpolation method has been briefly discussed in section 4-1, more explanation will be needed to enter the cubic B-spline with the theory of curves.

Faux (1979) deduced a formula of the cubic B-spline curve segment which may be written in the form :-

$$\mathbf{r}(u) = \frac{1}{6}\begin{bmatrix}1 & u & u^2 & u^3\end{bmatrix}\begin{Bmatrix}1 & 4 & 1 & 0\\ -3 & 0 & 3 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{Bmatrix}\begin{Bmatrix}\mathbf{P}_1\\ \mathbf{P}_2\\ \mathbf{P}_3\\ \mathbf{P}_4\end{Bmatrix} \qquad (4\text{-}11)$$

for $0 \le u \le 1$

and for short $\mathbf{r}(u) = \{u\}[A]\{\mathbf{P}_i\}$ only for $0 \le u \le 1$

where $\mathbf{P}_i$ are the position vectors of the control polygon points, as shown in Fig.(4-4).

The relation between the curve segment and its control points can be written from the equation (4-11) by using the end condition of the segment u=0 and u=1 as follows;

$$r(u=0)= \frac{2}{3}P_2 + \frac{1}{6}\left[P_1+P_3\right] \quad (4\text{-}12)$$

$$r(u=1)= \frac{2}{3}P_3 + \frac{1}{6}\left[P_2+P_4\right] \quad (4\text{-}13)$$

The cubic basis functions of the B-spline curve segments is all non negative on (0,1), and sum to unity. Then any point of segment is a weighted average of the four control points defining the associated polygon. This implies that the entire curve segment lies within the convex hull of its control points, therefore with complex curves more development must be needed for this function.

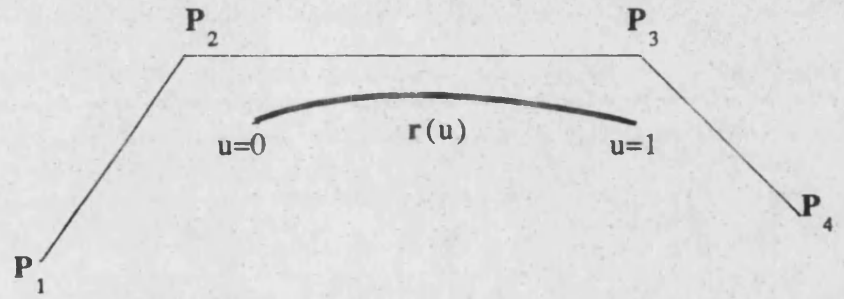


Fig.(4-4) The cubic B-spline curve segment

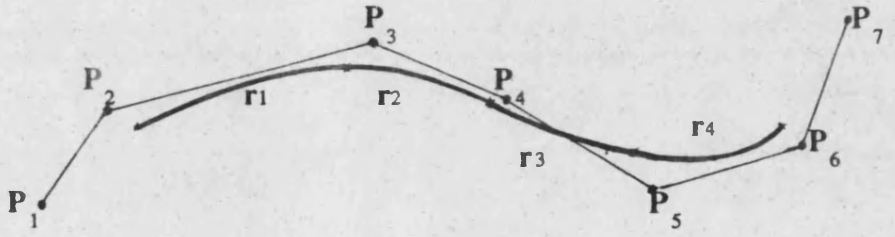
4-2-2 Composite B-spline curve

The single parametric cubic segments are not capable to represent the complex curves which will be used in the applications later in chapter 5. However, such curves can be synthesized as sequences of cubic segments. It is important to verify the continuity of the curves up to the second degree, as clarified by Pratt, that means the tangent and the curvature must be continuous, this allows more freedom in the construction of

smooth composite curves.

Therefore with the composite cubic B-spline curve, this continuity can be obtained automatically by arranging successive control polygons to overlap as shown in Fig.(4-5). In this figure it can be seen each successive control polygon has three of its four points in common with the preceding one.

According to the cubic B-spline curve segment, the values $\mathbf{r}_1(u=1)$ and $\mathbf{r}_2(u=0)$ coincide up to their second derivatives, so $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$ are continuous across the joint.



Fig(4-5) Composite cubic B-spline curve

4-2-3 cubic B-splines

Expansion of the right-hand side of equation (4-11) reveals that the cubic B-spline curve segment on the interval $0 \le u \le 1$ is defined in terms of the set of basis functions;

$$\frac{1}{6}(1-u)^3 \ , \ \frac{1}{6}(4-6u^2+3u^3) \ , \ \frac{1}{6}(1+3u+3u^2-3u^3) \ , \ \frac{1}{6}u^3$$

By considering a composite B-spline curve defined by a set of control points $[P_1, P_2, \ldots, P_n]$ as shown in Fig.(4-5), the second segment of the curve is expressed by equation (4-11) with the control points in the column vector are replaced by P_2, P_3, P_4, P_5 and u replaced by u_2, therefore the local parameter runs from 0 to 1 on the new segment. However, by setting $u_2 = u-1$ where u runs from 1 to 2 on the second segment, the parameter u can be used as a single global parameter over both segments. This stratagem allows parameterisation of the entire curve in terms of u, which runs from 0 to n over the n segments and is related to the local parameter u_i on the i^{th} segment by $u_i = u-i+1$.

The curves representing these functions are illustrated on the relevant interval in Fig.(4-6)

In terms of the global parameter u the basis functions on the i segment are translated a distance i-1 along the u-axis as shown in the Fig.(4-6). From this figure it's clear that the basis functions on each integer interval in u is continuous with those on either adjacent interval.

In fact an analysis based on equation (4-11) shows that the second continuity is obtained between the curve segments where they join at integer value of u. The bold curve labelled $N_3(u)$, in Fig.(4-6), is a second degree continuous curve composed of four cubic segments with the u-axis at u=0 and u=4, so that the continuity of this curve is verified for all u but which departs from 0 to 4 only on a four intervals.

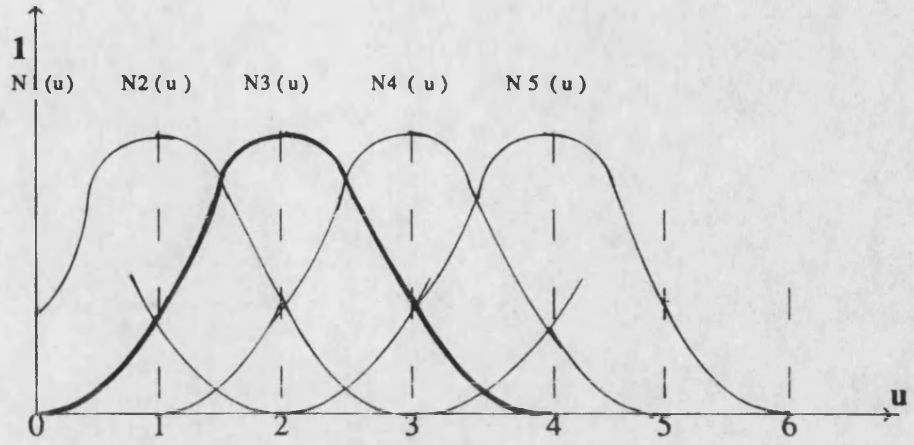


Fig.(4-6) Uniform cubic B-spline functions

4-3 Parametric cubic surfaces

4-3-1 The bicubic B-spline patch

Pratt (1984) specified the bicubic B-spline patch analogous to the cubic B-spline curve segment which has been defined in section 4-2. The equation of the patch is given by;

$$\mathbf{r}(u,v)=\begin{pmatrix}1 & u & u^2 & u^3\end{pmatrix}DTD^T\begin{pmatrix}1 & v & v^2 & v^3\end{pmatrix}$$

only $0 \le u \le 1$ and $0 \le v \le 1$ (4-14)

where u and v are parameters as shown in Fig.(4-7),

and $[D]=\frac{1}{6}\begin{Bmatrix}1 & 4 & 1 & 0\\ -3 & 0 & 3 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{Bmatrix}$

The central matrix [T] contains the coefficients of the basis functions. These coefficients are the position vectors of a 4X4 array of space points which form the vertices of a control polyhedron of the surface patch.

$$T=\begin{Bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{Bmatrix} \tag{4-15}$$

However, the relation between the patch and the polyhedron is different from the B-spline case, as shown in Fig.(4-7).

The figure shows that none of the control points lies on the surface in general. Also, the B-spline surface patch lies inside the convex hull of its control polyhedron.

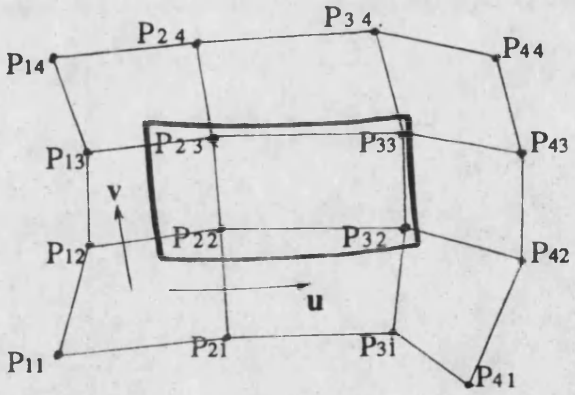


Fig.(4-7) A bicubic B-spline patch and its control polyhedron

4-3-2 Composite Bicubic Surface

The main condition of the composite surface is to keep the continuity in all components of **r**(u,v) as functions of u and v across the boundary. Then, the continuity of the tangent and curvature of the surface is usually required.

Resulting from the overlapping of the control polyhedra of neighbouring patches, the bicubic B-spline surface has automatic continuity of the second degree, as shown in Fig.(4-8) at least 12 control points overlap between each two adjoining surface patch sharing a common boundary curve. Moreover, because the B-spline surface has a local modification property similar to that of the B-spline curve; a change in one control point modifies a maximum of sixteen patches of the composite surface, leaving the remainder undisturbed.

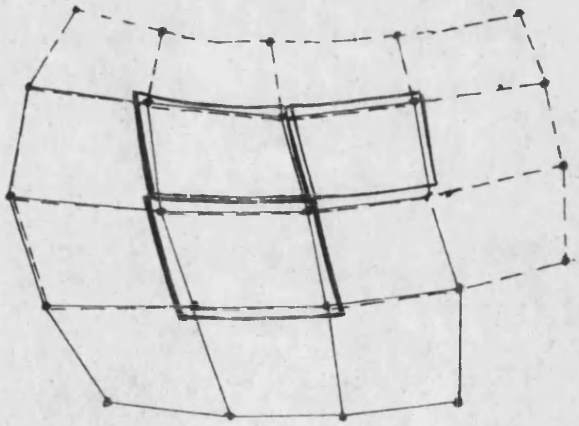


Fig.(4-8) Part of the Control polyhedron
of a Bicubic B-spline Surface

There are many studies that have been carried out to produce smooth surfaces subjected to these boundaries and conditions, such as Pratt (1984), Sturge (1986), Gordon (1974) and Williams (1980) ,(1986). These studies can be used in the design of car bodies and aircraft as well as in the design of grid shells.

4-4 Curvilinear coordinates

The coordinates x^i or x_i referred to a right handed orthogonal cartesian system of axes define a three-dimensional Euclidean space. A general coordinates θ_i or θ^j can be introduced by the transformation;

$$\theta^i = \theta^i(x_1,x_2,x_3) \quad \text{or} \quad \theta^j = \theta^j(x_1,x_2,x_3) \tag{4-16}$$

where;

i and j =1,2,3 and θ_i are arbitrary single valued functions of cartesian coordinates x_i which can derivatives up to any required order and which are independent of each other. By reversing the transformation of the above equation, the x_i can be written as;

$$x_i = x_i(\theta^1,\theta^2,\theta^3) \quad \text{and} \quad x^j = x^j(\theta_1,\theta_2,\theta_3) \tag{4-17}$$

where the functions in (4-17) are also single-valued. With the above assumptions, each set of values of x_i corresponds to a unique set of values of θ_i and vice versa. Hence the variables θ_i determine points in the defined three-dimensional Euclidean space,

and it can represent the space instead of the cartesian system x_i.

The functions in equations (4-16),(4-17) are assumed to be single-valued and to have continuous derivatives so that the correspondence between (x^1,x^2,x^3) and $(\theta_1,\theta_2,\theta_3)$ is unique.

The transformation of differential $d\theta_i$ and dx^j are also unique and a linear one where;

$$d\theta_i = \frac{\partial x^j}{\partial \theta^i} dx_j \ . \qquad (4\text{-}18)$$

While the transformation of the variables x^j in (4-17) is not linear in general.

The relation $\theta_i(x^1,x^2,x^3)$ = constant is the equation of a surface and as the value of the constant varies a family of surfaces can be obtained. Also corresponding to i=1,2,3 three families of surfaces are created and the point of intersection of one member from each of these families determine one point in the space. The conditions imposed on the functions θ_i ensure that the three surfaces obtained by taking a member of each family intersect in one and only one point, thus defining the position of the point uniquely.

At this point, let R be the position vector which can be expressed as a function in coordinates θ_i thus

$$R=R(\theta_1,\theta_2,\theta_3) \qquad (4\text{-}19)$$

and from (4-18) and (4-19);

$$d\mathbf{R}=\mathbf{g}_i d\theta^i = \mathbf{g}^i d\theta_i \qquad (4\text{-}20)$$

where,

$\mathbf{g}_1,\mathbf{g}_2,\mathbf{g}_3$ and $\mathbf{g}^1,\mathbf{g}^2,\mathbf{g}^3$ are the covariant the contrvariant base vectors as mentioned in Green and Zerna.

The relation between the two kinds of the base vector can be seen in Fig. (4-9).

The surfaces θ_i= constant is called coordinate surfaces and refer to them briefly as the θ_i-surfaces. The intersections of these surfaces also give three curves through every point P, two of them lying on each coordinate surface. These curves are called coordinate curves, and θ_i are called curvilinear coordinates.

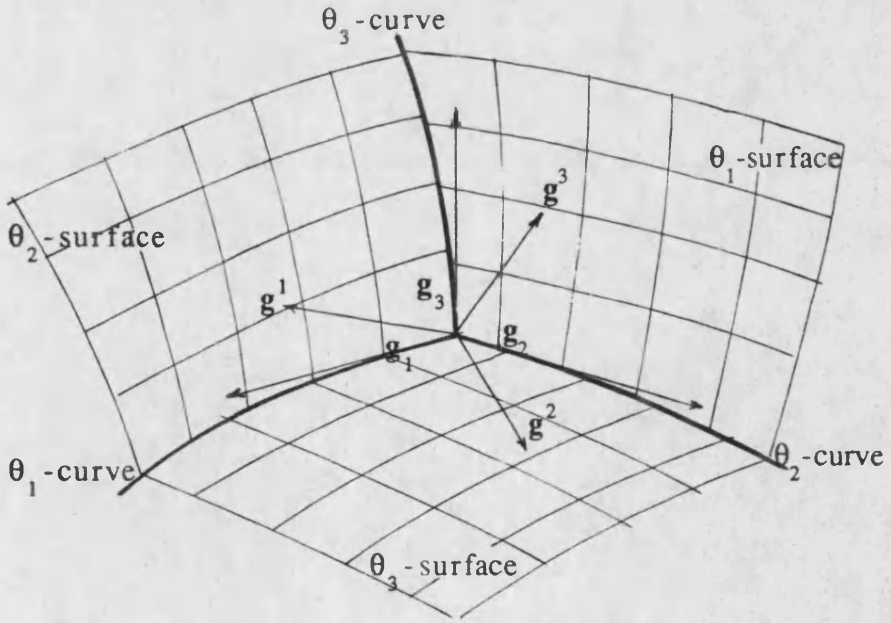


Fig.(4-9)

4-5 Introduction to differential geometry of curves

Classical differential geometry is the mathematical study of the general properties of curves and surfaces embedded in three dimensional Euclidean space. There are many authors like Struik (1961), Chung (1988), Coxeter (1961) and Lipschutz (1969) whom defined the curve in different ways. For instance Struik, illustrated the curve in space as a path of a point in motion, and by using the parametric formulation, the equation of the curve in space was given in the form;

$$x_i = x_i(\theta) \qquad \theta_1 \leq \theta \leq \theta_2 \qquad (4\text{-}21)$$

The parametric geometry is conventionally used in the computer aided design of free-form curves and surfaces, but the main advantage of using the parametric formulation is that any points on curves or surfaces can be easily computed.

On the other hand Coxeter studied the concept of the curve using the vector analysis as shown in Fig.(4-10). The Figure shows that the curve C is given by expressing the radius vector OP=r of a generic point P as;

$$\mathbf{r} = x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \qquad (4\text{-}22)$$

where $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are the unit vectors in positive direction of the coordinate x_1, x_2 and x_3.

The unit tangent of the curve C at point P may be calculated from the first derivative of **r** with respect to the arc length of the curve, s, as

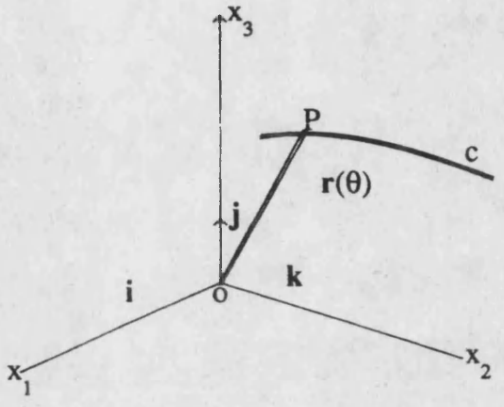


Fig.(4-10)

$$\mathbf{t}(s) = \mathbf{r}'(s) = \frac{d\mathbf{r}}{ds} \tag{4-23}$$

If another parameter θ is used instead of, s, the unit tangent can be written (with respect to θ) as follows,

$$\mathbf{t}(\theta) = \frac{\mathbf{r}'(\theta)}{|\mathbf{r}'(\theta)|} \tag{4-24}$$

In fact, using a natural parameter like θ is considered more general and convenient especial during the calculation of the surface's quantities as it will be seen in section (4-7).

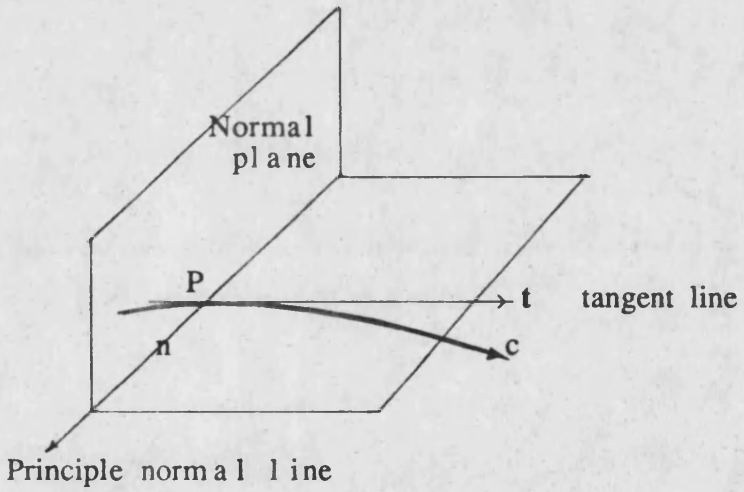
By differentiating the unit tangent $\mathbf{t}(\theta)$ with respect to θ,

the unit normal vector **n**, which is perpendicular to the tangent, can be obtained. Thus the curvature can be calculated as follows,

$$\frac{d\mathbf{t}}{d\theta} = \mathbf{K} = \dot{\mathbf{t}} = \mathbf{t}' = k\,\mathbf{n} \qquad (4\text{-}25)$$

The value of k being positive or negative is according to the direction of **n** whether is on the concave or convex side of the curve as shown in Fig.(4-11).

From the above two orthogonal and continuous unit vectors **t** and **n**, another unit vector **b** is called the unit binormal vector is obtained from the vector product of **t** and **n** as follows,



Fig(4-11)

$$\mathbf{b} = \mathbf{t} \times \mathbf{n} \qquad (4\text{-}26)$$

Therefore the three orthogonal vectors **n**,**t** and **b** can form a

three orthogonal plane as shown in Fig.(4-12). From equation (4-26) the second curvature or the torsion can be calculated by using the first derivative of that equation,

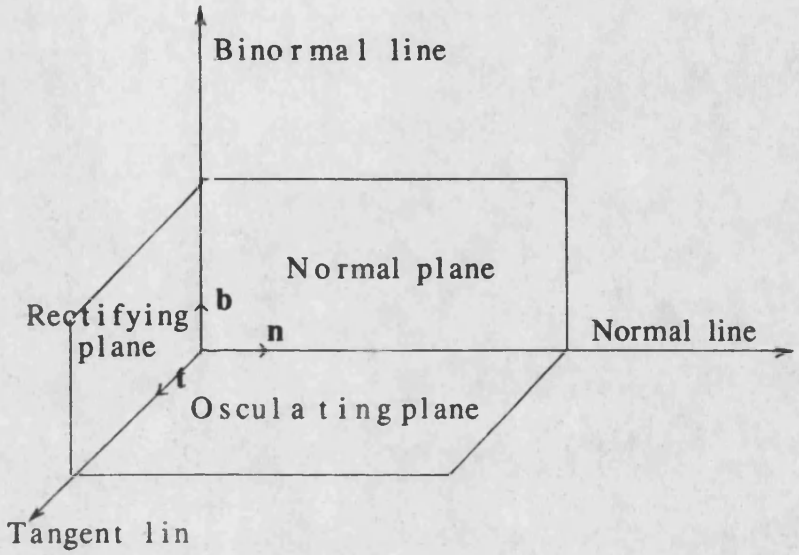


Fig.(4-12)

$$\dot{\mathbf{b}} = \dot{\mathbf{t}} \times \mathbf{n} + \mathbf{t} \times \dot{\mathbf{n}}$$

$$\dot{\mathbf{b}} = k\left(\mathbf{n} \times \mathbf{n} \right) + \mathbf{t} \times \dot{\mathbf{n}}$$

$$\dot{\mathbf{b}} = \quad 0 \quad + \mathbf{t} \times \dot{\mathbf{n}} \qquad (4\text{-}27)$$

also, $$\dot{\mathbf{n}} = \mu\mathbf{t} + \tau\mathbf{b} \qquad (4\text{-}28)$$

where $\dot{\mathbf{n}}$ is orthogonal to $\mathbf{n}$ and therefore is parallel to the rectifying plane so that, $\dot{\mathbf{n}}$ is a linear combination of $\mathbf{t}$ and $\mathbf{b}$ as shown, by substituting equation (4-28) in equation (4-27).

$$\dot{\mathbf{b}} = \mathbf{t} \times \left(\mu.\mathbf{t} + \tau.\mathbf{b} \right) = \tau\left(\mathbf{t} \times \mathbf{b} \right) = - \tau\mathbf{n} \qquad (4\text{-}29)$$

where the **t,n** and **b** are right handed orthogonal triplet and **txb=-n** then the torsion equation can take this final form,

$$\tau = -\dot{\mathbf{b}}.\mathbf{n} \qquad (4\text{-}30)$$

Therefore the regular curve is uniquely determined by those two scalar quantities, curvature and torsion, as a function of the natural parameter from equations (4-25) and (4-30) respectively.

4-6 The parametric formulation of the surfaces

The tensor analysis is widely used in differential geometry and numerous problems in three dimensional analysis. The main advantage of the tensor analysis that it is easy to transform a set of coordinates from one frame of reference to another (see Green and Zerna (1968), or Bickley and Gibson (1965).

The well known physical quantities such as scalars (like mass, volume) and vectors (like velocity, force, etc..) are considered the simplest quantities that can be described by tensor notation. In three dimensions a vector is represented by three quantities, its components referred to some coordinate system. It is rather more difficult to describe some entities as vectors, for instance, the component of a stress distribution is associated with two directions. Therefore, use of the tensor analysis made it easy to seek expressions for such entities which are formally the same for all coordinate systems.

For instance; the scalar quantity is called tensor of rank zero,

the vector quantity is tensor of rank one,

the stress distribution is tensor of rank two,

and there are even more complicated quantities which are called tensors of higher rank.

The geometric study of a surface (θ_1,θ_2) embedded in three dimensions (x^1,x^2,x^3) may be shown in Fig.(4-13) which represent a portion of surface that define the position vector,

$$\mathbf{r}=\mathbf{r}(\theta_1,\theta_2)$$

$$=x^1(\theta_1,\theta_2)\,\mathbf{i}_1+x^2(\theta_1,\theta_2)\mathbf{i}_2+x^3(\theta_1,\theta_2)\mathbf{i}_3 \qquad (4\text{-}31)$$

or $$=x^k\mathbf{i}_k \quad \text{for short where } k=1,2 \text{ and } 3.$$

where x^k are cartesian coordinates and $\mathbf{i}_k$ unit vectors in the direction of increasing x^k. Vectors will always be denoted by bold type.

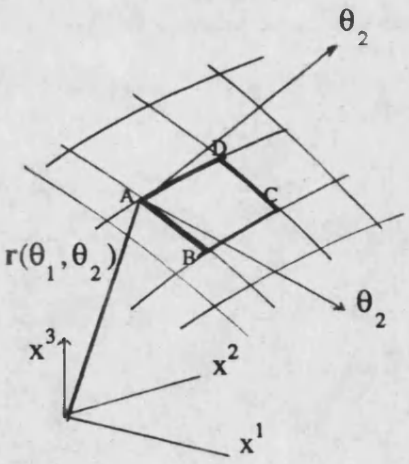


Fig.(4-13)

At each point on the surface there are two sets of base vectors. Firstly there are the covariant base vectors

$$\mathbf{a}_\alpha = \frac{\partial \mathbf{r}}{\partial \theta^\alpha} \tag{4-32}$$

where, α means partial differentiation with respect to θ^α and may have the value of 1 or 2.

In general $\mathbf{a}_1$ and $\mathbf{a}_2$ are not unit vectors and they lie in the local plane of the surface in the direction of increasing θ^1 and θ^2 respectively. The third covariant base vector, $\mathbf{a}_3$, is the local unit normal to the surface.

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \mathrm{x} \mathbf{a}_2}{|\mathbf{a}_1 \mathrm{x} \mathbf{a}_2|} \tag{4-33}$$

The two contravariant base vectors, $\mathbf{a}^1$ and $\mathbf{a}^2$, lie in the local plane of the surface where $\mathbf{a}^1$ is perpendicular to $\mathbf{a}_2$ and $\mathbf{a}^2$ is perpendicular to $\mathbf{a}_1$, and have direction and magnitude such that the scalar product

$$\mathbf{a}^\alpha . \mathbf{a}_\beta = \delta^\alpha_\beta \quad = 1 \text{ if } \alpha=\beta \qquad = 0 \text{ if } \alpha\neq\beta \tag{4-34}$$

where δ^β_α are the Kronecker deltas.

The third contravariant, $\mathbf{a}^3$, is also the unit normal so that;

$$\mathbf{a}^3 = \mathbf{a}_3 \tag{4-35}$$

The surface is usually known by a set of points in space which resembles a portion of a plane in the nighbourhood of each of its points, and this surface is considered the image of a sufficiently regular mapping of a set of points in the space.

Struik (1961) expressed the equation of the surface by its rectangular coordinates x^i, as a function of two parameters θ_1 and θ_2 in a certain closed interval;

$$x^i=x^i(\theta_1,\theta_2) \qquad (4\text{-}36)$$

where $\theta_1^1 \le\theta_1\le\theta_1^2$ and $\theta_2^1\le\theta_2\le\theta_2^2$

Lipschutz (1969) introduced the surface equation using the vectors analysis, then equation (4-36) may be written again as follows;

$$x_i^i=x_i^i(\theta_1,\theta_2)=x^1g_1+ x^2g_2+ x^3g_3 \qquad (4\text{-}37)$$

The parameters θ_1 and θ_2 must enter independently in the surface equation.

By using the θ_1 and θ_2 Parameters, the surface can be formed first by keeping θ_2 constant, then x will depend only on one parameter, θ_1, and thus determines a curve on the surface (a parametric curve) θ_2=constant. Secondly, θ_1=constant represents another parametric curve. When the constants vary, the surface is covered with a net of parametric curves, two of which pass through every point P, forming the family of ∞ curves θ_2=constant and the family of ∞ curves θ_1=constant.

The geometry of surfaces determined by certain local invariant quantities is called the first and second fundamental forms and can be referred to those forms as follows;

4-6-1 First fundamental form of the surface

The tensor notation is useful in dealing with the surface analysis, in sense, it makes the subject more compact, but at the same time more elusory. It might be more easier to understand this form using the classical notation as in Struik. In spite of that the tensor analysis will be used in the rest of this chapter which will be used in chapter 6. Green and Zerna (1968) provided a full explanation of using the tensor analysis in surface, thus tensor notation can be used to write those forms as follows,

The second order tensor $\mathbf{a}^{\alpha\beta}$ and $\mathbf{a}_{\alpha\beta}$ defined by the scalar products;

$$\mathbf{a}_{\alpha\beta} = \mathbf{a}_{\beta\alpha} = \mathbf{a}_{\alpha}.\mathbf{a}_{\beta}$$

and

$$\mathbf{a}^{\alpha\beta} = \mathbf{a}^{\beta\alpha} = \mathbf{a}^{\alpha}.\ \mathbf{a}^{\beta} \qquad (4\text{-}38)$$

are the covariant and contravariant metric surface tensors respectively. The distance, (ds) between two adjacent points on the surface is given by,

$$\begin{aligned} ds^2 &= dr.dr \\ &= a_{\alpha} d\theta^{\alpha}.a_{\beta} d\theta^{\beta} \\ &= a_{\alpha\beta}\ d\theta^{\alpha} d\theta^{\beta} \\ &= a_{11}(d\theta^1)^2 + 2a_{12}(d\theta^1)(d\theta^2) + a_{22}(d\theta^2)^2 \end{aligned} \qquad (4\text{-}39)$$

This equation is known the first fundamental form of the surface.

4-6-2 Second fundamental form of the surface

The second order tensor $b_{\alpha\beta}$ can be defined from the scalar product of dr and da_3 as follows;

$$dr.da_3=-a_\alpha.a_{3,\beta}=a_3.a_{\alpha,\beta}=a_3.a_{\beta,\alpha}$$

$$=-b_{\alpha\beta}\, d\theta^\alpha d\theta^\beta=-b^{\alpha\beta}\, d\theta_\alpha d\theta_\beta=-b^\alpha_\beta\, d\theta^\alpha d\theta^\beta \qquad (4\text{-}40)$$

then,

$$b_{\alpha\beta} = b_{\beta\alpha} =-a_\alpha.a_{3,\beta}=-a_\beta.a_{3,\alpha}=a_3.a_{\alpha,\beta}=a_3.a_{\beta,\alpha} \qquad (4\text{-}41)$$

where $a_\alpha.a_3=0$

The subscript $\alpha\beta$ means partial differentiation with respect to θ^α and θ^β.

$b_{\alpha\beta}$ is a symmetric second order tensor.

a_3 is a unit vector and thus $a_{3,\beta}$ lies in the plane of the surface.

In undergoing a small displacement, (dr), on the surface the unit normal will change by (da_3). The component of (da_3) in the direction of (dr) is due to the normal curvature in that direction and the component of (da_3) perpendicular to (dr) is due to the twist of the surface in the direction of (dr). then the scalar product

$$dr.da_3 = (a_\alpha d\theta^\alpha).(-b_{\mu\beta}.a^\mu d\theta^\beta)$$

$$= -b_{\alpha\beta}\, d\theta^\alpha d\theta^\beta \qquad (4\text{-}42)$$

is known as the second fundamental form of the surface and the normal curvature in the direction of dr is equal to

$$-\frac{dr.da_3}{dr.dr} = -\frac{\text{the second fundamental form}}{\text{the first fundamental form}} \tag{4-43}$$

$$= \frac{b_{\alpha\beta}\, d\theta^{\alpha} d\theta^{\beta}}{a_{\lambda\rho} d\theta^{\lambda} d\theta^{\rho}} \tag{4-44}$$

The minimum and maximum values of the normal curvature are the principal curvatures, k_1, k_2 which always occur in orthogonal directions. The mean curvature, H, is the mean of the two values of k;

$$2H = b^{\alpha}_{\alpha} = \frac{a_{11}b_{22} - 2a_{12}b_{12} + a_{22}b_{11}}{a_{11}a_{22} - (a_{12})^2} \tag{4-45}$$

and the Gaussian curvature, G, is the product of the two values;

$$G = \frac{b_{11}b_{22} - (b_{12})^2}{a_{11}a_{22} - (a_{12})^2} = b^1_1 b^2_2 - b^2_1 b^1_2 = \frac{1}{2}(b^{\alpha}_{\alpha} b^{\beta}_{\beta} - b^{\beta}_{\alpha} b^{\alpha}_{\beta}) \tag{4-46}$$

The Christoffel symbols of the second kind can be defined as;

$$\Gamma^{\lambda}_{\alpha\beta} = \Gamma^{\lambda}_{\beta\alpha} = a^{\lambda}.a_{\alpha,\beta} \tag{4-47}$$

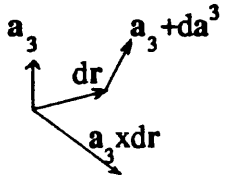
The expressions for the derivatives of the base vectors of the surface, are written in the formulae of Weingarton and Gauss as;

$$a_{\alpha,\beta} = \Gamma^{\lambda}_{\alpha\beta}\, a_{\lambda} + b_{\alpha\beta}\, a_3$$

$$a^{\alpha}{}_{,\beta} = -\Gamma^{\alpha}_{\beta\lambda}\, a^{\lambda} + b^{\alpha}_{\beta}\, a_3$$

$$a_{3,\alpha} = -\, b^{\lambda}_{\alpha}\, a_{\lambda} \tag{4-48}$$

The twist is calculated as ;



$$\text{Twist}= \frac{(a_3 x dr).da^3}{dr.dr} \qquad (4\text{-}49)$$

From the second fundamental form and Gaussian curvature, Gauss theorem can be produced. Gauss theorem is particularly important since it effectively shows that the Gaussian curvature can be expressed in terms of the metric tensor (a_{11},a_{22},a_{12}) and its derivatives only as;

$$G= \frac{b_{11}b_{22}-(b_{12})^2}{a_{11}a_{22}-(a_{12})^2} = \frac{1}{a_{11}a_{22}-(a_{12})^2}\left[a_{12,12}- \frac{1}{2}(a_{11,22}+a_{22,11}) -\Gamma^{\lambda}_{11}\Gamma^{\rho}_{22}a_{\lambda\rho}+ \Gamma^{\lambda}_{12}\Gamma^{\rho}_{12}a_{\lambda\rho}\right] \qquad (4\text{-}50)$$

(calculation of this equation is shown in Struik (1961) and Williams (1987)).

In the cable structures and grid shells, the Gaussian curvature depends on the angle α between the crossed elements, then equation (4-50) is written as;

$$G = \frac{b_{11}b_{22}-(b_{12})^2}{1-(a_{12})^2} = - \frac{1}{\sin \alpha} \frac{\partial^2 \alpha}{\partial\theta^1\partial\theta^2} \qquad (4\text{-}51)$$

where $a_{11}=a_{22}=1$.

PART II

INTRODUCTION TO PART II

Part I of the thesis discussed the buckling behaviour of structures and the analytical and numerical methods used in buckling analysis.

In part II of the thesis, this knowledge will be applied to the buckling of grid shells. However before studying the grid shell properly the analytical and numerical techniques are first studied with a plane curve.

Chapter seven discusses an experimental study on a grid shell model. A general conclusions and possible suggestions for future research on grid shells is written in chapter eight.

CHAPTER FIVE

STRAIN ENERGY EQUATION OF A PLANE CURVE USING CUBIC B-SPLINE FUNCTION

5-1 General

The finite element analysis is applied to calculate the buckling load of a plane curve using the cubic B-spline function. The analysis studied the bending and stretching stresses of the curve as well as the boundary conditions.

5-2 The total potential energy of a plane curve

From the principle of minimum total potential energy which has been mentioned in part 1, the strain energy equation of a deformed curve C with length ds due to bending is

$$Q_1 = \int_0^{ds} \text{Moment} \times \text{curvature}\ ds \ . \qquad (5\text{-}1)$$

or

$$= \frac{1}{2} \int_0^{ds} \beta\ (\text{curvature})^2\ ds \ . \qquad (5\text{-}2)$$

where β is the bending stiffness of the curve.

From Fig.(5-1) the curvature of the curve C is $= \frac{d\psi}{ds}$ then equation (5-2) is written as follows;

$$Q_1 = \frac{\beta}{2} \int_0^{ds} \left[\frac{d\psi}{ds}\right]^2 ds \ . \qquad (5\text{-}3)$$

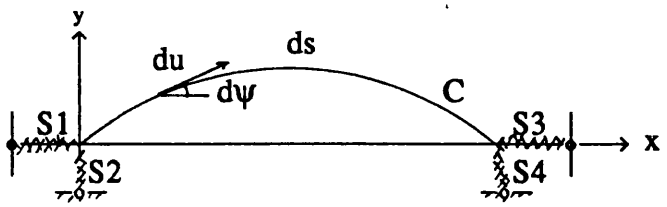


Fig.(5-1)

The strain energy equation of the curve due to stretching is

$$Q_2 = \int_0^{ds} \text{Stress x strain ds} . \qquad (5\text{-}4)$$

or

$$= \frac{1}{2} \int_0^{ds} \alpha \, (\text{strain})^2 \, ds . \qquad (5\text{-}5)$$

where α is the axial stiffness of the curve, and from Fig.(5-1) this equation can be written as follows,

$$Q_2 = \frac{\alpha}{2} \int_0^{ds} \left[\frac{\frac{1}{2}\left[\left(\frac{ds}{du}\right)_f^2 - \left(\frac{ds}{du}\right)_i^2\right]}{\left(\frac{ds}{du}\right)_i^2} \right]^2 ds . \qquad (5\text{-}6)$$

where $\left(\frac{ds}{du}\right)_f$ and $\left(\frac{ds}{du}\right)_i$ are the final and initial curve length respectively. Also the strain energy of the springs S1,S2,S3 and S4 is

$$Q_3 = \frac{1}{2} \Sigma \, S.\delta^2 . \qquad (5\text{-}7)$$

where, S and δ are the spring's stiffness and displacement respectively.

The potential energy of the external forces P in the deformed configuration is defined as;

$$Q_4 = -\sum P.\Delta \qquad (5\text{-}8)$$

where Δ is the displacement due to the force P .

From the above equations, the total potential energy equation of the plane curve is equal to $Q_1+Q_2+Q_3+Q_4$.

Hence,

$$I = \frac{\beta}{2}\int_0^{ds}\left[\frac{d\psi}{ds}\right]^2 ds + \frac{\alpha}{2}\int_0^{ds}\left[\frac{\frac{1}{2}\left[\left(\frac{ds}{du}\right)_f^2 - \left(\frac{ds}{du}\right)_i^2\right]}{\left(\frac{ds}{du}\right)_i^2}\right]^2 ds + \frac{1}{2}\Sigma S\delta^2$$

$$- \sum P.\Delta \qquad (5\text{-}9)$$

This equation can be written as a function in a parameter du by multiplying the equation in $\left(\frac{ds}{du}\,\frac{du}{ds}\right)$ then;

$$I = \frac{\beta}{2}\int_{u=0}^{u=1}\left(\frac{d\psi}{du}\right)^2 \frac{du}{ds}.du + \frac{\alpha}{2}\int_{u=0}^{u=1}\left[\frac{\frac{1}{2}\left[\left(\frac{ds}{du}\right)_f^2 - \left(\frac{ds}{du}\right)_i^2\right]}{\left(\frac{ds}{du}\right)_i^2}\right]^2 \frac{ds}{du}.du$$

$$+ \frac{1}{2}\Sigma S\,\delta^2 - \sum P.\Delta \qquad (5\text{-}10)$$

and by substituting $\frac{ds}{du}=L$ therefore, equation (5-10) can be written as follows;

$$I = \frac{\beta}{2}\int_{u=0}^{u=1}\left(\frac{d\psi}{du}\right)^2 \frac{1}{L}\,du + \frac{\alpha}{8}\int_{u=0}^{u=1} L\left[\frac{\left[\mathbf{r}'(u).\mathbf{r}'(u)\right]_f}{L^2} - 1\right]^2 du$$

$$+ \frac{1}{2}S.\delta^2 - \sum P.\Delta \qquad (5\text{-}11)$$

where $\left(\frac{ds}{du}\right)_f^2 = \left[\mathbf{r}'(u).\mathbf{r}'(u)\right]_f$

Substituting the co-ordinates of the control points of cubic B-spline curve (which has been explained in section 4-2), the first and second derivatives can be calculated with respect to u as shown;

$$\mathbf{r}'(u) = \frac{1}{6}\,[0 \quad 1 \quad 2u \quad 3u^2] \begin{bmatrix} & & & \\ & & A & \\ & & & \end{bmatrix} [\mathbf{P}] \qquad (5\text{-}12)$$

and for short,

$$\mathbf{r}'(u) = U' \left[A \right] \left\{ \mathbf{P} \right\}$$

where $U' = [0 \quad 1 \quad 2u \quad 3u^2]$.

Also;

$$\mathbf{r}''(u) = \frac{1}{6}\,[0 \quad 0 \quad 2 \quad 6u] \begin{bmatrix} & & & \\ & & A & \\ & & & \end{bmatrix} [\mathbf{P}] \qquad (5\text{-}13)$$

and for short,

$$\mathbf{r}''(u) = U'' \left[A \right] \left\{ \mathbf{P} \right\}$$

where $U'' = [0 \quad 0 \quad 2 \quad 6u]$

$$\left\{ \mathbf{P} \right\} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \\ \mathbf{P}_4 \end{bmatrix} = \begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ x_3+y_3 \\ x_4+y_4 \end{bmatrix}$$

as shown in Fig.(5-2).

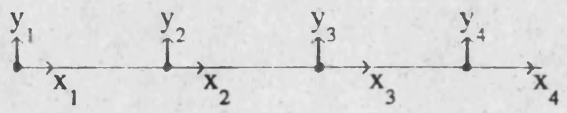


Fig.(5-2)

To carry out the solution of equation (5-11), the equivalent load and stiffness matrix for each term can be calculated individually as it will be explained .

5-2-1 Calculation of the bending force and stiffness (Q_1)

$$\text{The normal curvature } \left(\frac{d\psi}{du}\right) = \left| \left(\mathbf{r}''(u).\mathbf{r}''(u) \right) \right| \qquad (5\text{-}14)$$

$$\text{hence, } \int_{u=0}^{u=1} \left(\frac{d\psi}{du}\right)^2 .du = \int_{u=0}^{u=1} \left(\mathbf{r}''(u).\mathbf{r}''(u) \right) du \qquad (5\text{-}15)$$

Substituting by equation (5-14), the above form is written as follows,

$$\int_{u=0}^{u=1} \left(\mathbf{r}''(u).\mathbf{r}''(u) \right) du = \int_{u=0}^{u=1} \{P\}^T [A]^T U''^T .U'' [A]\{P\} \qquad (5\text{-}16)$$

$$\text{So, assume } [C2] = U''^T .U'' = \begin{Bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 12u \\ 0 & 0 & 12u & 36u^2 \end{Bmatrix} \qquad (5\text{-}17)$$

from the last substitution in equation (5-17), the middle part is assumed to be,

$$[D] = [A]^T [C2] [A] \qquad (5\text{-}18)$$

$$\left[D \right]=\frac{1}{36}\begin{pmatrix}1 & -3 & 3 & -1\\ 4 & 0 & -6 & 3\\ 1 & 3 & 3 & -3\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 12u\\ 0 & 0 & 0 & 36u^2\end{pmatrix}\begin{pmatrix}1 & 4 & 1 & 0\\ -3 & 0 & 3 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{pmatrix} \tag{5-19}$$

$$\left[D \right] = \begin{Bmatrix} 1-2u+u^2 & -2+5u-3u^2 & 1-4u+3u^2 & u-u^2 \\ & 4-12u+9u^2 & -2+9u-9u^2 & -2u^2+3u^3 \\ & & 1-6u+9u^2 & u-3u^2 \\ \text{(symmetric)} & & & u^2 \end{Bmatrix} \tag{5-20}$$

Then the strain energy of the plane curve due to bending is,

$$Q_1=\frac{\beta}{2}\int_{u=0}^{u=1}\frac{1}{L}\left(r''(u).r''(u) \right)du=\frac{\beta}{2}\int_{u=0}^{u=1}\frac{1}{L}\left\{ P \right\}^T\left[D \right]\left\{ P \right\}du \tag{5-21}$$

The integration of equation (5-21) ,with respect to the co-ordinates of the control points, will end to a new matrix $\left[D_I\right]$ containing the integrated value of [D], for u varying from 0 to 1 as shown,

$$Q_1=\frac{\beta}{2L}\left\{ P \right\}^T\left[D_I \right]\left\{ P \right\}\Big|_0^1 \tag{5-22}$$

where;

$$[D_I]=\begin{bmatrix} u-u^2+\frac{u^2}{3} & -2u+\frac{5}{2}u^2-u^3 & u-2u^2+u^3 & \frac{u^2}{2}-\frac{u^3}{3} \\ & 4u-6u^2+3u^3 & -2u+\frac{9}{2}u^2-3u^3 & -u^2+u^3 \\ & & u-3u^2+3u^3 & \frac{u^2}{2}-u^3 \\ & & & \frac{u^3}{3} \end{bmatrix} \qquad (5\text{-}23)$$

The equivalent load on the control points in the x direction is equal to $\left\{\frac{\partial Q_I}{\partial x_i}\right\}$.

The equivalent load on the control points in the y direction is equal to $\left\{\frac{\partial Q_I}{\partial y_i}\right\}$

where,

$$\left\{\frac{\partial Q_I}{\partial x_i}\right\}^T = \left\{\frac{\partial Q_I}{\partial x_1} \quad \frac{\partial Q_I}{\partial x_2} \quad \frac{\partial Q_I}{\partial x_3} \quad \frac{\partial Q_I}{\partial x_4}\right\} \qquad (5\text{-}24)$$

$$\left\{\frac{\partial Q_I}{\partial y_i}\right\}^T = \left\{\frac{\partial Q_I}{\partial y_1} \quad \frac{\partial Q_I}{\partial y_2} \quad \frac{\partial Q_I}{\partial y_3} \quad \frac{\partial Q_I}{\partial y_4}\right\} \qquad (5\text{-}25)$$

$$\frac{\delta Q_I}{\delta x_1} = \frac{\beta}{L}[i \quad 0 \quad 0 \quad 0]\begin{bmatrix} & & & \\ & D_I & & \\ & & & \end{bmatrix}\begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ x_3+y_3 \\ x_4+y_4 \end{bmatrix} \qquad (5\text{-}26)$$

$$\frac{\delta Q_I}{\delta y_1} = \frac{\beta}{L}[j \quad 0 \quad 0 \quad 0]\begin{bmatrix} & & & \\ & D_I & & \\ & & & \end{bmatrix}\begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ x_3+y_3 \\ x_4+y_4 \end{bmatrix} \qquad (5\text{-}27)$$

in which,

i and **j** are the unit vector in x and y directions respectively.

Hence,

$$\frac{\partial Q_1}{\partial x_1} = \frac{\beta}{L}\left\{D_{I(1,1)}*x_1 + 0 + D_{I(1,2)}*x_2 + 0 + D_{I(1,3)}*x_3 + 0 + D_{I(1,4)}*x_4 + 0\right\}$$

$$\frac{\partial Q_1}{\partial y_1} = \frac{\beta}{L}\left\{0 + D_{I(1,1)}*y_1 + 0 + D_{I(1,2)}*y_2 + 0 + D_{I(1,3)}*y_3 + 0 + D_{I(1,4)}*y_4\right\}$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\frac{\partial Q_1}{\partial y_4} = \frac{\beta}{L}\left\{0 + D_{I(1,1)}*y_1 + 0 + D_{I(1,2)}*y_2 + 0 + D_{I(1,3)}*y_3 + 0 + D_{I(1,4)}*y_4\right\}.$$

(5-28)

So, the stiffness matrix for normal bending is

$$\left(\frac{\partial^2 Q_1}{\partial x_i \partial y_j}\right) \qquad \text{where i and j=1 to 4} \qquad (5\text{-}29)$$

and

$$\frac{\partial^2 Q_1}{\partial x_1 \partial y_j} = \frac{\beta}{L}\left\{\frac{\partial^2 Q_1}{\partial x_1 \partial y_1} \quad 0 \quad \frac{\partial^2 Q_1}{\partial x_1 \partial y_2} \quad 0 \quad \frac{\partial^2 Q_1}{\partial x_1 \partial y_3} \quad 0 \quad \frac{\partial^2 Q_1}{\partial x_1 \partial y_4} \quad 0\right\}$$

$$\frac{\partial^2 Q_1}{\partial x_i \partial y_1} = \frac{\beta}{L}\left\{0 \quad \frac{\partial^2 Q_1}{\partial x_1 \partial y_1} \quad 0 \quad \frac{\partial^2 Q_1}{\partial x_2 \partial y_1} \quad 0 \quad \frac{\partial^2 Q_1}{\partial x_3 \partial y_1} \quad 0 \quad \frac{\partial^2 Q_1}{\partial x_4 \partial y_1}\right\}.$$

The normal bending stiffness matrix= KBE(8x8) can take the form

$$\frac{\beta}{L}\begin{bmatrix}
D_{I(1,1)} & 0 & D_{I(1,2)} & 0 & D_{I(1,3)} & 0 & D_{I(1,4)} & 0 \\
0 & D_{I(1,1)} & 0 & D_{I(1,2)} & 0 & D_{I(1,3)} & 0 & D_{I(1,4)} \\
D_{I(2,1)} & 0 & D_{I(2,2)} & 0 & D_{I(2,3)} & 0 & D_{I(2,4)} & 0 \\
0 & D_{I(2,1)} & 0 & D_{I(2,2)} & 0 & D_{I(2,3)} & 0 & D_{I(2,4)} \\
D_{I(3,1)} & 0 & D_{I(3,2)} & 0 & D_{I(3,3)} & 0 & D_{I(3,4)} & 0 \\
0 & D_{I(3,1)} & 0 & D_{I(3,2)} & 0 & D_{I(3,3)} & 0 & D_{I(3,4)} \\
D_{I(4,1)} & 0 & D_{I(4,2)} & 0 & D_{I(4,3)} & 0 & D_{I(4,4)} & 0 \\
0 & D_{I(4,1)} & 0 & D_{I(4,2)} & 0 & D_{I(4,3)} & 0 & D_{I(4,4)}
\end{bmatrix}$$

(5-30)

where the values of $D_{I(i,j)}$ are shown in Appendix (B-1-a).

5-2-2 Calculation of the axial force and stiffness(Q_2)

The calculation of the second term in equation (5-11) can be carried out by studying the axial strain value which can be obtained from the following formula;

$$St=\frac{1}{2}\left[\frac{\mathbf{r}'(u).\mathbf{r}'(u)}{L^2}-1\right]^2. \qquad (5\text{-}31)$$

From equation (5-12), the scalar product of $\left(\mathbf{r}'(u).\mathbf{r}'(u)\right)$ may be calculated as follows;

$$\mathbf{r}'(u).\mathbf{r}'(u)=\left\{P\right\}^T\left[A\right]^T\mathbf{U}'^T.\mathbf{U}'\left[A\right]\left\{P\right\}. \qquad (5\text{-}32)$$

To simplify the previous equation then,

$$\left[C_1\right]=\mathbf{U}'^T.\mathbf{U}'=\begin{Bmatrix}0&0&0&0\\0&1&2u&3u^2\\0&2u&4u^2&6u^3\\0&3u^2&6u^3&9u^4\end{Bmatrix}. \qquad (5\text{-}33)$$

Substituting equation (5-33), the middle part of equation (5-32) is

$$\left[B\right]=\left[A\right]^T\left[C_1\right]\left[A\right] \qquad (5\text{-}34)$$

therefore,

$$\left[B\right]=\frac{1}{36}\begin{Bmatrix}1&-3&3&-1\\4&0&-6&3\\1&3&3&-3\\0&0&0&1\end{Bmatrix}\begin{Bmatrix}0&0&0&0\\0&1&2u&3u^2\\0&2u&4u^2&6u^3\\0&3u^2&6u^3&9u^4\end{Bmatrix}\begin{Bmatrix}1&4&1&0\\-3&0&3&0\\3&-6&3&0\\-1&3&-3&1\end{Bmatrix} \qquad (5\text{-}35)$$

$$\left[B \right] = \frac{1}{4} \begin{Bmatrix} B(1,1) & B(1,2) & B(1,3) & B(1,4) \\ & B(2,2) & B(2,3) & B(2,4) \\ & & B(3,3) & B(3,4) \\ & \text{symmetric} & & B(4,4) \end{Bmatrix} \qquad (5\text{-}36)$$

where the values of the matrix [B] are in Appendix (B-1-b).

Then,

$$r'(u).r'(u) = \left\{ P \right\}^T \left[B \right] \left\{ P \right\} . \qquad (5\text{-}37)$$

Using equation (5-37), the value of Q_2 is

$$Q_2 = \frac{\alpha L}{8} \left[\frac{1}{L^4} \int_{u=0}^{u=1} \left\{ P \right\}^T \left[B \right] \left\{ P \right\} \left\{ P \right\}^T \left[B \right] \left\{ P \right\} du \right.$$

$$\left. - \frac{2}{L^2} \int_{u=0}^{u=1} \left\{ P \right\}^T \left[B \right] \left\{ P \right\} du \right] . \qquad (5\text{-}38)$$

The first part of this equation can be solved by using integration by parts as shown in the following procedure.

This method usually needs to calculate the differentiation and integration of the variable many times, where in this case the integration will be carried out up to the fifth integration for matrix [B] with respect to u degree. So, equation (5-38) can be rewritten as;

$$Q_2 = \frac{\alpha L}{8} \left[\frac{1}{L^4} \left(\{P\}^T[BI_1]\{P\}\{P\}^T[B]\{P\} - \{P\}^T[BI_2]\{P\}\ \{P\}^T[BD_1]\{P\} \right.\right.$$

$$+ \{P\}^T[BI_3]\{P\}\{P\}^T[BD_2]\{P\} - \{P\}^T[BI_2]\{P\}\ \{P\}^T[BD_3]\{P\}$$

$$\left.\left. + \{P\}^T[BI_5]\{P\}\{P\}^T[BD_4]\{P\} \right) - \frac{2}{L^2} \{P\}^T[BI_1]\{P\} \right] \qquad (5\text{-}39)$$

where, BI_1, BI_2, BI_3, BI_4 and BI_5 are the integration values of [B] up to the fifth integration.

BD_1, BD_2, BD_3 and BD_4 are the derivatives [B] up to the fourth derivative.

The equivalent load of the control points can be calculated from the differentiation of equation (5-39) with respect to the co-ordinates of the control point as follows;

$$\left\{ \frac{\partial Q_2}{\partial x_i} , \frac{\partial Q_2}{\partial y_i} \right\} \qquad \text{where} \quad i=1 \text{ to } 4 \tag{5-40}$$

$$\text{where} \quad \left\{ \frac{\partial Q_2}{\partial x_i} \frac{\partial Q_2}{\partial y_i} \right\} = \left\{ \frac{\partial Q_2}{\partial x_1} \frac{\partial Q_2}{\partial y_1} \frac{\partial Q_2}{\partial x_2} \frac{\partial Q_2}{\partial y_2} \frac{\partial Q_2}{\partial x_3} \frac{\partial Q_2}{\partial y_3} \frac{\partial Q_2}{\partial x_4} \frac{\partial Q_2}{\partial y_4} \right\}^T \tag{5-41}$$

$$\text{and the differentiation of the} \quad \left(P \right)_1 = \frac{\partial P}{\partial x_1} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} . \tag{5-42}$$

The values of the $\frac{\partial Q_2}{\partial x_1}, \frac{\partial Q_2}{\partial y_1}, \frac{\partial Q_2}{\partial x_2}, \frac{\partial Q_2}{\partial y_2}, \frac{\partial Q_2}{\partial x_3}, \frac{\partial Q_2}{\partial y_3}, \frac{\partial Q_2}{\partial x_4}$ and $\frac{\partial Q_2}{\partial y_4}$ are shown in Appendix (B-1-c).

The stiffness values can also be calculated by using the second differentiation with respect to the co-ordinates of the control point of the equation (4-39) as follows;

$$KAE(i,j) = \frac{\partial Q_2{}^2}{\partial X_i \; \partial Y_j} \tag{4-43}$$

and those values are shown in Appendix (B-1-d).

5-2-3 Calculation of equivalent load and stiffness of the spring supports (Q_3)

In the analysis of the plane curve, springs will be used at a certain position as shown in Fig.(5-3). The stiffness, S, of the springs and its position are defined in the data.

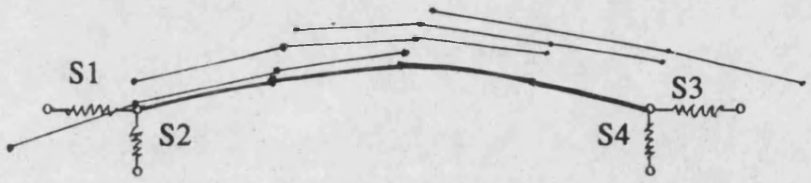


Fig.(5-3)

Equivalent load and stiffness values, can be calculate from the first and the second derivatives of equation (5-7) which represents the strain energy of the spring.

$$PS_i = \frac{\partial Q_3}{\partial x_i} = S_i.\delta_i \frac{\partial \delta_i}{\partial x_i}. \tag{5-44}$$

$$KS_i = S_i.\frac{\partial \delta_i}{\partial x_i}.\frac{\partial \delta_i}{\partial y_j} + S_i.\delta_i.\frac{\partial^2 \delta_i}{\partial x_i \partial y_j}. \tag{5-45}$$

To simplify the solution of the equation.(5-45) the second part of this equation is very small, so it can be neglected, then

$$KS_i = S_i.\frac{\partial \delta_i}{\partial x_i}.\frac{\partial \delta_i}{\partial y_i} \tag{5-46}$$

where, δ_i= (x or y component of $\mathbf{r}(u)$) - constant (5-47)

and the constant is the initial value of the spring position as shown in Fig.(5-4).

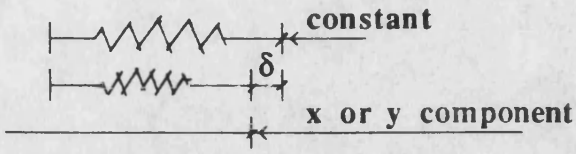


Fig.(5-4)

Therefore, equations (5-44) and (5-46) can be written as follows;

$$PS(i)=S_i\left[\left(D(i,j).x_i - \text{constant}\right).D(i,j)\right] \qquad (5\text{-}48)$$

$$KS(i,j)=S_i.D(i,j).D(j,i) \qquad (5\text{-}49)$$

where the values of D(i,j) were defined in equation (5-20)

5-2-4 Calculation of potential energy of the external force Q_4

The potential energy of the applied load is = -P.Δ so, the equivalent load on the control points is

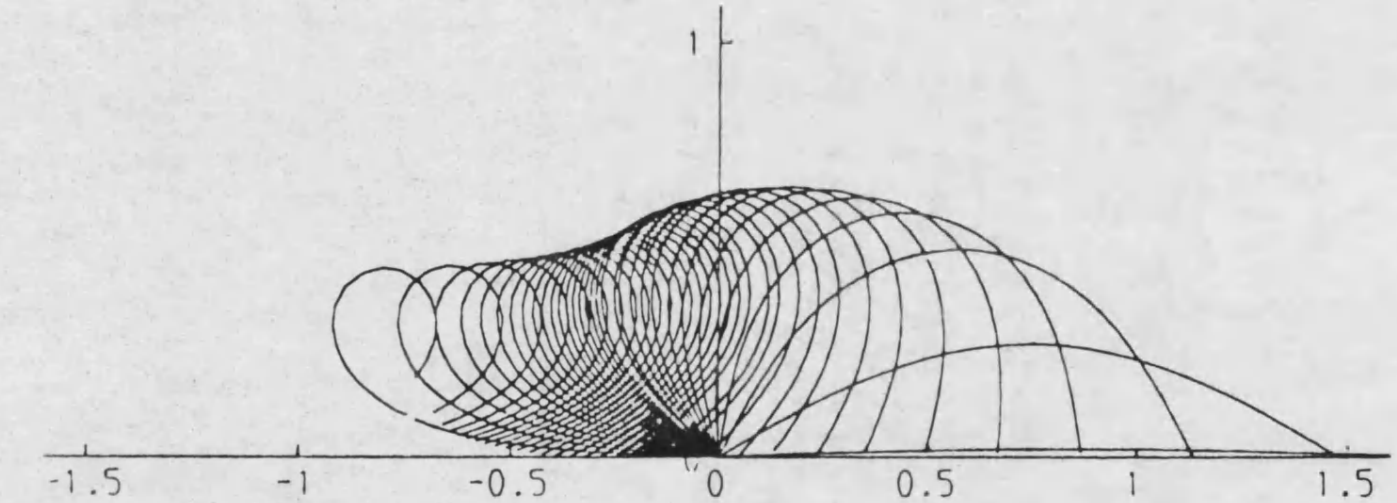
$$\frac{\partial \Delta}{\partial x_i} = \sum_i \frac{\partial Q_4}{\partial x}$$ where i = No. of degree of freedom.

Hence,

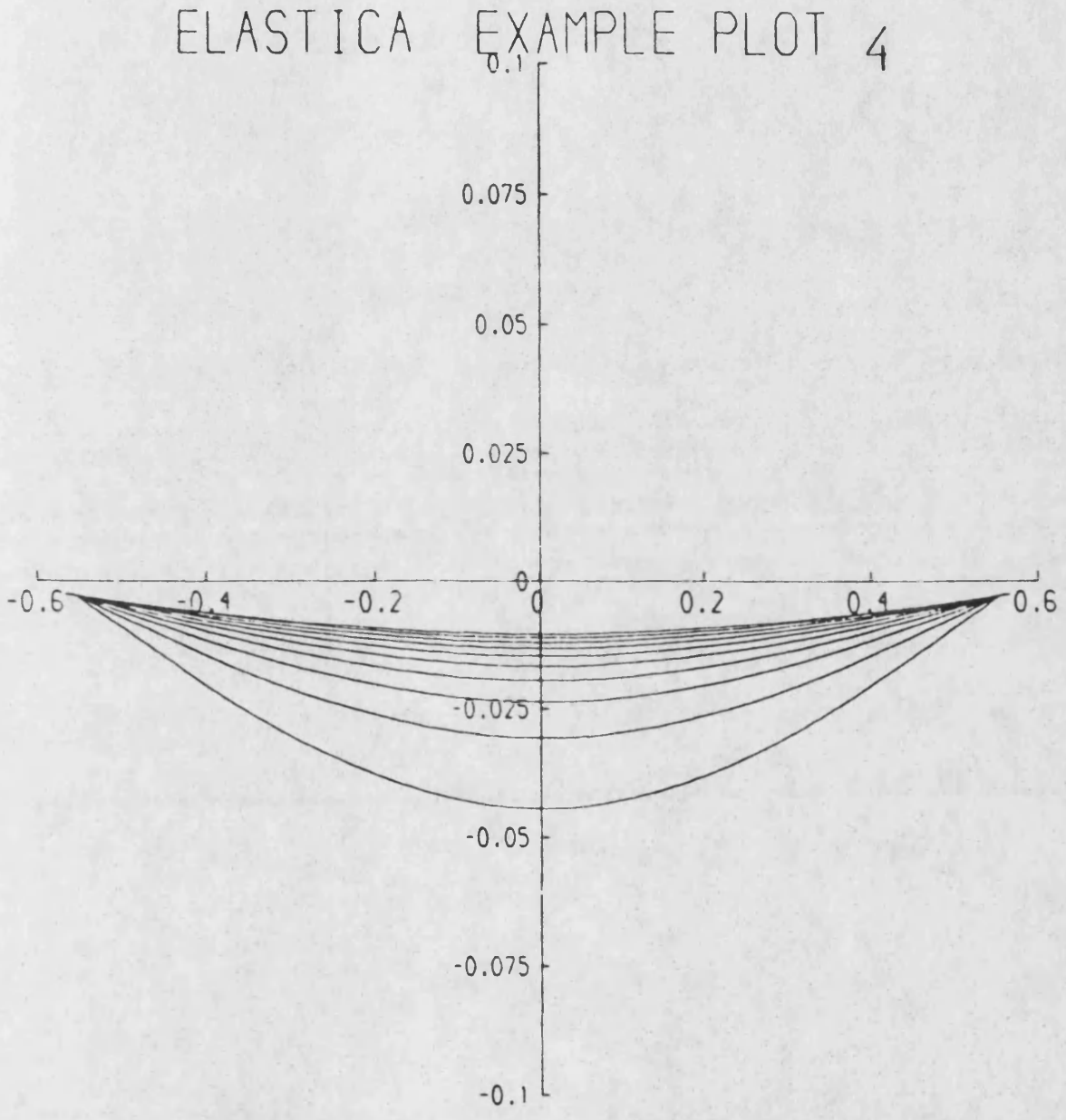
$$FP(j) = P.\frac{\partial \Delta}{\partial x_i} = P.D(1,j) \qquad (5\text{-}50)$$

From the above analysis a computer program was written, using the finite element analysis and the cubic B-spline function, to

calculate the buckling load of the plane curve. The flow-diagram is shown in Appendix (B-2), Fig(5-5) shows the behaviour of the curve under an axial load, and Fig.(5-6) shows the curve under intermediate load.



Fig(5-5) Column subjected to axial force (Output from 2Dprog.)



Fig(5-6) Beam subjected to intermediate load (Output from 2Dprog)

5-3 Theory of Elastica

The theory of the elastica is studied to verify the results which was obtained from the analysis of the finite element computer program.

The problem of the elastica was first solved by Euler in 1744, the theory assumes no plastic deformation takes place (see Karman 1940). Therefore this assumption would only apply to extremely slender columns as shown in Fig. (5-7). The exact solution of the problem requires a solution of a non linear differential equation which determines the shape of the deflected central line of the column as follows,

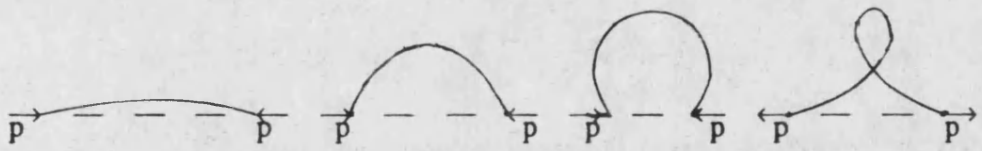


Fig.(5-7) Progressive deformation of an elastic column

$$\frac{d^2y}{dx^2} + \frac{py}{EI}\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{3/2} = 0 \quad \text{(see equation (1-1))}$$

By assuming small deflection and neglecting the second order term $\left(\frac{dy}{dx}\right)^2$, this simplification leads to a simple second order linear differential equation of the form;

$$\frac{d^2y}{dx^2} + \frac{p}{EI} y = 0 \qquad (5\text{-}51)$$

$$\text{or} \quad Py = -EI \frac{d^2y}{dx^2} = -\beta \frac{d\psi}{ds} \qquad (5\text{-}52)$$

where β=EI is the bending stiffness of the curve(s) and its tangent make an angle ψ with the horizontal axis as shown in the Fig. (5-8).

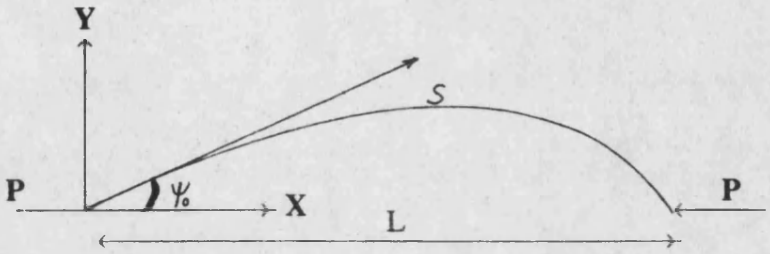


Fig.(5-8)

By assuming $y=\int \sin\psi \; ds$

so, $$y' = \frac{\partial y}{\partial s} = \sin\psi \qquad (5\text{-}53)$$

by differentiating both sides of equation (5-52) and substituting in equation (5-53), the following equation can be produced,

$$P \sin\psi = -\beta \frac{d^2\psi}{ds^2} \qquad (5\text{-}54)$$

by multiplying the above equation by $\left(\frac{d\psi}{ds}\right)$ and integration then;

$$-P \cos\psi = -\frac{1}{2}\beta \left(\frac{d\psi}{ds}\right)^2 + P.C \qquad (5\text{-}55)$$

where C is a constant,

$$\frac{\beta}{2P}\left(\frac{d\psi}{ds}\right)^2 = \cos\psi + C$$

$$\sqrt{\frac{2P}{\beta}}\int ds = \int \frac{d\psi}{\sqrt{(\cos\psi + C)}} \qquad (5\text{-}56)$$

equation (5-56) is considered as an elliptic integral and to simplify the solution the numerical methods can be used as follows;

$$\int ds = \sqrt{\frac{\beta}{2P}} \int_{-\psi_o}^{+\psi_o} \frac{d\psi}{\sqrt{(\cos\psi + C)}} \qquad (5\text{-}57)$$

where at the end of the curve y=0;

$\psi = \pm\psi_o$ and also $\frac{d\psi}{ds} = 0$

$$\frac{\beta}{2P}\left(\frac{d\psi}{ds}\right)^2 = (\cos\psi - \cos\psi_o)$$

$$S = -\sqrt{\frac{\beta}{2P}} \int_{-\psi_o}^{+\psi_o} \frac{d\psi}{\sqrt{(\cos\psi - \cos\psi_o)}} \qquad (5\text{-}58)$$

where the (-) sign is with the square root ,therefore the full arc length can be calculated as the following,

$$L = \sqrt{\frac{\beta}{2P}} \int_{-\psi_o}^{+\psi_o} \frac{d\psi}{\sqrt{(\cos\psi + \cos\psi_o)}} \qquad (5\text{-}59)$$

because of $\cos\psi = \left(1\text{-}2\sin^2\frac{\psi}{2}\right)$

and $\cos\psi_o = \left(1\text{-}2\sin^2\frac{\psi_o}{2}\right)$

then by substituting those values in equations, (5-59) the arc length of the curve is;

$$L = \sqrt{\frac{\beta}{2P}} \int_{-\psi_o}^{+\psi_o} \frac{d\psi}{\sqrt{2\left(\sin^2\frac{\psi_o}{2} - \sin^2\frac{\psi_o}{2}\right)}} \qquad (5\text{-}60)$$

$$L = \sqrt{\frac{\beta}{2P}} \left\{\frac{1}{\sin\frac{\psi_o}{2}}\right\} \int_{-\psi_o}^{+\psi_o} \frac{d\psi}{\sqrt{2\left(1-(\sin^2\frac{\psi}{2}/\sin^2\frac{\psi_o}{2})\right)}} \qquad (5\text{-}61)$$

$$L = \sqrt{\frac{\beta}{P}} \left\{\frac{1}{\sin\frac{\psi_o}{2}}\right\} \int_{0}^{\psi_o} \frac{d\psi}{\sqrt{\left(1-(\sin^2\frac{\psi}{2}/\sin^2\frac{\psi_o}{2})\right)}}. \qquad (5\text{-}62)$$

By making some transformation to solve the above problems numerically, suppose $\sin\frac{\psi}{2} = x$ and after diff. this value,

$$\frac{1}{2}\cos\frac{\psi}{2}\, d\psi = dx$$

and $$d\psi = 2\,\frac{dx}{\cos\frac{\psi}{2}} = 2\,\frac{dx}{\sqrt{\left(1-\sin^2\frac{\psi}{2}\right)}} = 2\,\frac{dx}{\sqrt{(1-x^2)}}$$

by substituting in equation (5-62) where $\sin\frac{\psi}{2}=x$ and $\sin\frac{\psi_o}{2}=0$

then;

$$\sqrt{\frac{PL^2}{\beta}} = \frac{1}{a}\int_{a}^{0} \frac{a.\,\frac{2dx}{\sqrt{(1-x^2)}}}{\sqrt{(a^2-x^2)}} = \int_{a}^{0} \frac{\frac{2dx}{\sqrt{(1-x^2)}}}{\sqrt{(a-x)(a+x)}} \qquad (5\text{-}63)$$

the second transformation can be made by assuming $\frac{1}{a-x}=y$

so $\frac{1}{y}=a-x$ or $x=a-\frac{1}{y}=\frac{ay-1}{y}$

thus, $dy=\frac{1}{(a-x)^2}dx$ or $dx=(a-x)^2dy=\frac{1}{y^2}dy$

where, at x=a $y=+\infty$ and at x=0 $y=\frac{1}{a}$

from the above relation the final shape of equation (5-63) can be written as follows

$$\sqrt{\frac{PL^2}{\beta}}=\int_{+\infty}^{1/a}\frac{2}{\sqrt{(2ay-1).[y^2-(ay-1)^2]}}dy \qquad (5\text{-}64)$$

where the $+\infty$ should be a very big value corresponding to (1/a), then by using trial and error a satisfactory accuracy can be achieved. By using numerical integration (like trapezoidal rule).

$$\int f(x)dx=\frac{h}{2}\left[f(x_o)+f(x_1)\right]-\frac{h^3}{12}f^2(\varepsilon) \qquad \text{at } n=1 \qquad (5\text{-}65)$$

where $x_o<\varepsilon<x_1$ then the expression (1/a) can be calculated and from, a, the initial angle, ψ_o, can be known, therefore if a and p are known, the deflection buckling can be calculated for the axial loaded column. Fig. (5-9) shows the mode of the elastica with initial angle (ψ_o) and load increment, where Fig. (5-10) shows the behaviour of the elastica with different initial angle(ψ_o).

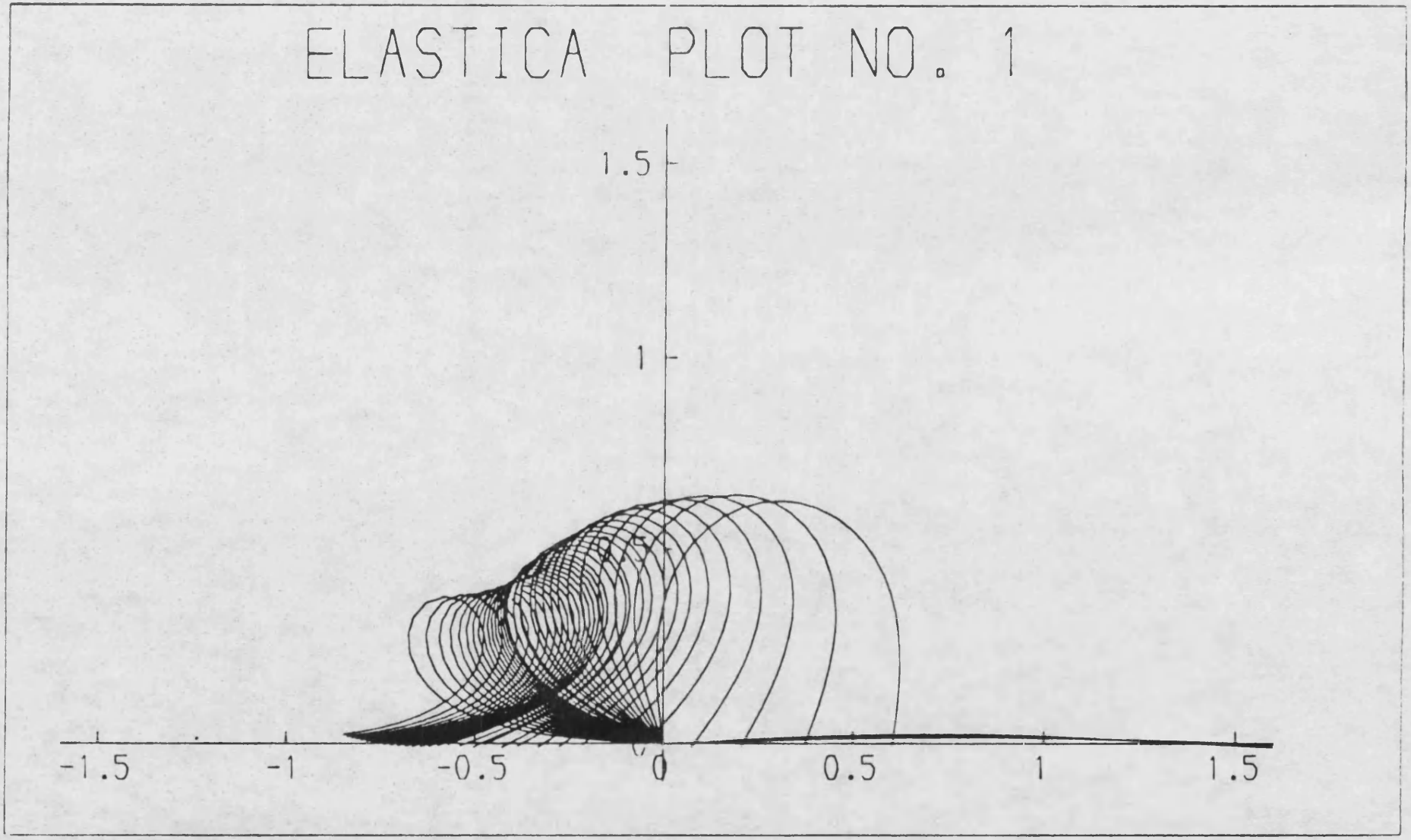


Fig.(5-9) The behaviour of elastica subjected to different loads

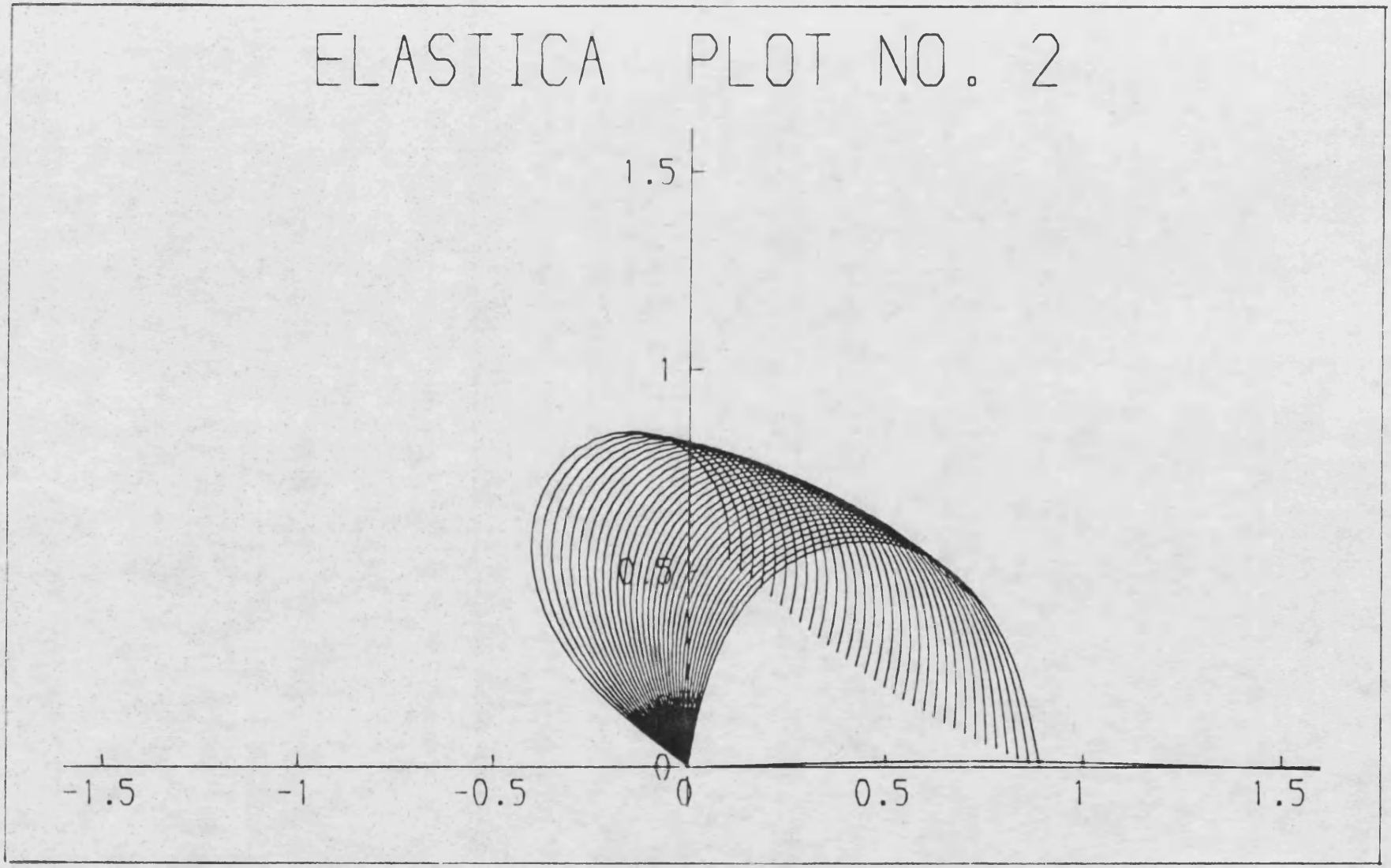


Fig.(5-10) The behaviour of elastica with different initial angles

5-4 Results and Conclusion

From the application on a plane curve in two dimensions, the deformed shape of the curve under different types of loading were predicted and a final conclusion can be written as follows;

1-The results obtained from the two-dimensional computer program are analogous with those obtained from the theory of elastica.

2-The cubic B-spline function can provide a smooth and continuous curve, also the deformation at any point on the curve can simply be calculated.

3-The shape and the type of the supports have a great influence on the behaviour of the curves.

The finite element analysis and cubic B-spline function can be extended to study the behaviour of the grid shell and predict the buckling load as it will be explained in chapter 6.

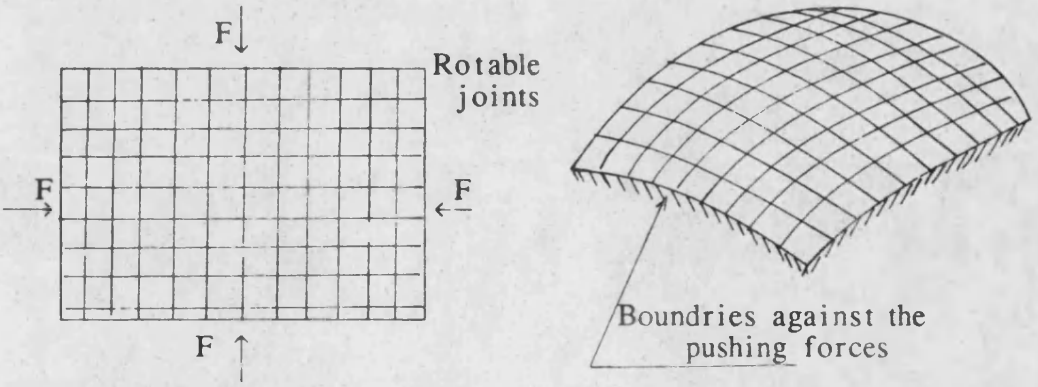
CHAPTER SIX

THREE DIMENSIONAL ANALYSIS OF GRID SHELLS USING VIRTUAL WORK EQUATION AND BICUBIC B-SPLINE FUNCTION

6-1 Introduction

The main aim of this chapter is to study the buckling load of grid shells subjected to external force by using the virtual work equation. The analysis includes the effect of the bending and torsional moment as well as the axial stretching on the behaviour of grid shells. The bicubic B-spline patch with 16 control points is used.

The erection method of the grid shell is usually started from a horizontal position, where the lattice is rotatable at the inter-section points. Then by pulling or pushing up the edges and fixing it at certain boundaries against collapse, the required shape can be obtained as shown in Fig.(6-1) .



a) The flat shape b)The required shape

Fig.(6-1)

6-2 Inextensional deformation in grid shells

The complete solution of any stress problems is usually combined with the determination of a corresponding deformation. According to the Hook's law the empirical relation between tension and strain is;

$$T=EA.\varepsilon \qquad (6\text{-}1)$$

where;

T,E,A and ε are the tension, Young's modulus, cross-sectional area and strain.

In certain problems, such as the deformation of arches, displacements are almost entirely due to bending deformation rather than axial strain. In this situation the actual value of EA does not influence the behaviour, provided that it is large enough to effectively prevent axial strain.

Mathematically it is more satisfactory to replace equation (6.1) by $\varepsilon=0$ and this means that the tension, T, has to be determined by the consideration of equilibrium.

A grid shell deforms mainly by changing the curvature of the grid members both out of the plane of the surface and in the plane of the surface, that is normal and geodesic curvature. The axial strain in the grid members are so low that the condition $\varepsilon=0$ will be used. Note if the axial stiffness of ties is much less than that of grid members, the condition $\varepsilon=0$ can not be used for ties.

Even though the assumption $\varepsilon=0$ is made for grid members, the hypothetical compatibility set used in the virtual work equation will use deformation modes which violate the condition $\varepsilon=0$.

The virtual work equation which represent the relation between the tension force and strain is;

$$\sum P_i.\delta_i = \sum T_j.\varepsilon_j.L_j \qquad (6\text{-}2)$$

where P_i is the actual external load,

T_j is the tension force (unknown),

δ_i and ε_j are the displacement and the virtual strain,

L_j is the member length.

when δ_n =1 and all the other degrees of freedom =0

The relation between the change in stress and the change in strain can be written as;

$$P_n = \sum T_j[R] \qquad (6\text{-}3)$$

$$\text{where } [R] = \left\{ \frac{\partial \varepsilon_j}{\partial \delta_n} L_j \right\} \qquad (6\text{-}4)$$

From these equations, the relation between the surface constraint and the deformation at the degrees of freedom is shown in the following matrix.

$$\begin{bmatrix} \partial P_{ij} - \partial W_{ij} \\ \hline \partial \varepsilon_{ij} \end{bmatrix} = \left[\begin{array}{c|c} K_{ik} - L_{ik} & R_{im} \\ \hline Q_{jk} & 0 \end{array}\right] \begin{bmatrix} \Delta_k \\ \hline \partial T_m \end{bmatrix} \qquad (6\text{-}5)$$

where $$[Q_{jk}]=[R_{im}]^T \qquad (6\text{-}6)$$

$$\partial \varepsilon_{ij} = [Q_{jk}]\ \Delta_k \qquad (6\text{-}7)$$

6-3 Virtual work equation of grid shells

The virtual work equation under this condition can be written in the following form;

$$\int \left(P_{ij}\partial u_{ij}\right) d\theta_1 d\theta_2 = \int \left(M_B \partial k_B + M_G \partial k_G + M_\tau \partial k_\tau\right) d\theta_1 d\theta_2 + S.\Delta + \int \left(T\ \partial \varepsilon\right) d\theta_1 d\theta_2 \qquad (6\text{-}8)$$

where

M_B and M_g are the bending moment about axis in plane of surface and bending moment about normal axis respectively

M_τ is the torsion moment,

k_B,k_G and τ are the normal curvature, geodesic curvature and twist respectively

T_j is the tension forces in the surface.

S= the spring stiffness.

BY assuming ∂u_{ij} the displacement at the control points is arbitrary displacement and equal to unity. Then, from the virtual work relation the following equation can be deduced;

$$\sum P_{ij}=\beta_1 \int_{\theta_1=0}^{1} \int_{\theta_2=0}^{1} k_B \cdot \left(\frac{dk_B}{d\delta_{ij}}\right) d\theta_1 d\theta_2 + \beta_2 \int_{\theta_1=0}^{1} \int_{\theta_2=0}^{1} k_G \cdot \left(\frac{dk_G}{d\delta_{ij}}\right) d\theta_1 d\theta_2$$

$$+\gamma \int_{\theta_1=0}^{1} \int_{\theta_2=0}^{1} \tau \cdot \left(\frac{d\tau}{d\delta_{ij}}\right) d\theta_1 d\theta_2 + S\Delta\left(\frac{d\Delta}{d\delta_{ij}}\right) + \Sigma_j R_{ij} \cdot T_j \qquad (6\text{-}9)$$

where,

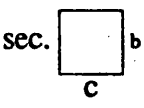
P_{ij} is the nodal load.

$\beta_1=EI_x$ (the bending stiffness in x direction)

$\beta_2=EI_y$ (the bending stiffness in y direction)

$\gamma=GJ$ where J is torsion factor and depends on member cross section as follows,

$J=\frac{\pi D^4}{32}$ for the circular section.

and $J=\frac{cb^3}{12}\left[4\text{-}2.52\frac{b}{2}+0.21\left(\frac{b}{c}\right)^5\right]$ for the Rect. sec. 

The first part of equation (6-9) which is concerned with the bending and torsional stress of the surface can be analyzed as follows;

The unit tangent in u-direction= $\frac{ru}{\sqrt{(ru \cdot ru)}}$

and in v-direction= $\frac{rv}{\sqrt{(rv.rv)}}$

the rate of change of unit tangent in u and v direction respectively is equal to;

$$\frac{\frac{\partial}{\partial u}\left[\frac{ru}{\sqrt{(ru.ru)}}\right]}{\sqrt{(ru.ru)}} \quad , \quad \frac{\frac{\partial}{\partial v}\left[\frac{rv}{\sqrt{(rv.rv)}}\right]}{\sqrt{(rv.rv)}} \qquad (6\text{-}10)$$

The change of normal curvature =

$$\left[\frac{\frac{\partial}{\partial u}\left[\frac{ru}{\sqrt{(ru.ru)}}\right]}{\sqrt{(ru.ru)}}\right].n + \left[\frac{\frac{\partial}{\partial v}\left[\frac{rv}{\sqrt{(rv.rv)}}\right]}{\sqrt{(rv.rv)}}\right].n \; . \qquad (6\text{-}11)$$

The change of geodesic curvature= $\left[\frac{\frac{\partial}{\partial u}\left[\frac{ru}{\sqrt{(ru.ru)}}\right]}{\sqrt{(ru.ru)}}\right] . \frac{(nXru)}{\sqrt{(ru.ru)}}$

$$+ \left[\frac{\frac{\partial}{\partial v}\left[\frac{rv}{\sqrt{(rv.rv)}}\right]}{\sqrt{(rv.rv)}}\right] . \frac{(nXrv)}{\sqrt{(rv.rv)}} . \qquad (6\text{-}12)$$

The change of twist = $\left[\frac{\frac{\partial}{\partial u}\left[\frac{nXru}{\sqrt{(ru.ru)}}\right]}{\sqrt{(ru.ru)}}\right].n$

$$+ \left[\frac{\frac{\partial}{\partial v}\left[\frac{nXrv}{\sqrt{(rv.rv)}}\right]}{\sqrt{(rv.rv)}}\right].n \; . \qquad (6\text{-}13)$$

where $\frac{\partial}{\partial u}\left[\frac{nXru}{\sqrt{(ru.ru)}}\right].n = -\frac{ru}{\sqrt{(ru.ru)}}.nu$

and $\frac{\partial}{\partial v}\left[\frac{nXrv}{\sqrt{(rv.rv)}}\right].n = -\frac{rv}{\sqrt{(rv.rv)}}.nv.$

From the above equations the general equation can be written in

the form;

$$F_{ij} = \frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_1\left[\frac{\partial\left[\frac{ru}{\sqrt{(ru.ru)}}\right]}{\sqrt{(ru.ru)}}\right]^2 .n\ d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_1\left[\frac{\partial\left[\frac{rv}{\sqrt{(rv.rv)}}\right]}{\sqrt{(rv.rv)}}\right]^2 .n\ d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\left[\frac{\partial\left[\frac{ru}{\sqrt{(ru.ru)}}\right]}{\sqrt{(ru.ru)}}\right]\left[\frac{nXru}{\sqrt{(ru.ru)}}\right]\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\left[\frac{\partial\left[\frac{rv}{\sqrt{(rv.rv)}}\right]}{\sqrt{(rv.rv)}}\right]\left[\frac{nXrv}{\sqrt{(rv.rv)}}\right]\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\gamma\left\{\left[\frac{\partial\left[\frac{nXru}{\sqrt{(ru.ru)}}\right]}{\sqrt{(ru.ru)}}\right].n\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\gamma\left\{\left[\frac{\partial\left[\frac{nXrv}{\sqrt{(rv.rv)}}\right]}{\sqrt{(rv.rv)}}\right].n\right\}^2 d\theta_1 d\theta_2 + \frac{1}{2}S\delta + P.\Delta \quad . \quad (6\text{-}14)$$

since $ru.n=0$

and $(nXru).n=0$

$$nuXru = -nXruu \qquad (6\text{-}15)$$

From the relations of equation (6-15), the general equation (6-14) can take the following simple form,

$$F_{ij}=\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_1.\left\{\frac{(ruu.n)}{(ru.ru)}\right\}^2 d\theta_1 d\theta_2 + \frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_1.\left\{\frac{(rvv.n)}{(rv.rv)}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2.\left\{\frac{ruu}{(ru.ru)}\frac{(nXru)}{\sqrt{(rv.rv)}}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2.\left\{\frac{rvv}{(rv.rv)}\frac{(nXrv)}{\sqrt{(rv.rv)}}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\gamma.\left\{-\frac{(nXru).nu}{(ru.ru)}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\gamma.\left\{-\frac{(nXrv).nv}{(rv.rv)}\right\}^2 d\theta_1 d\theta_2 + \frac{1}{2}S\delta + P.\Delta \ . \qquad (6\text{-}16)$$

6-3-1 Calculation of the normal bending term(Q_1)

The normal bending term (Q_1) can be written as follows (by using the tensor notation and by substituting in the first fundamental form of the surface $dr.dr=a_{\alpha b}\, d\theta_\alpha d\theta_\beta$);

$$Q_1=\frac{1}{2}\beta_1\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\left(\frac{a_{1,1}.a_3}{a_{11}}\right)^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\beta_2\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\left(\frac{a_{2,2}.a_3}{a_{22}}\right)^2 d\theta_1 d\theta_2 \qquad (6\text{-}17)$$

Using the relation $\mathbf{a}_{1,1}.\mathbf{a}_3=b_{11}$ and $\mathbf{a}_{2,2}.\mathbf{a}_3=b_{22}$ (the second fundamental forms of the surface) where b_{11} and b_{22} are symmetric surface tensors of order two, then equation (6-17) takes the form,

$$Q_1=\frac{1}{2}\beta_1\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\left(\frac{b_{11}}{a_{11}}\right)^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\beta_2\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\left(\frac{b_{22}}{a_{22}}\right)^2 d\theta_1 d\theta_2 \qquad (6\text{-}18)$$

where $\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_{1,1},\mathbf{a}_{2,2},a_{11},a_{22},b_{11},b_{22},b_{12}$ and $\mathbf{a}_3$ are given in Appendix (C-1).

The double integration of the surface along θ_1 and θ_2, in the above equation will be solved numerically by dividing the patch into small elements in both u and v directions. The value of the equivalent load on the control point and the stiffness term of the normal bending can be calculated from the first and second derivative of equation (6-18) as follows;

The equivalent load on the control point $FNB=\frac{\partial Q_1}{\partial \delta_{ij}}$

where,

$$\frac{\partial Q_1}{\partial \delta_{ij}}=\beta_1\left\{\left(\frac{b_{11}}{a_{11}}\right)\cdot\frac{\partial}{\partial\delta_{ij}}\left(\frac{b_{11}}{a_{11}}\right)+\left(\frac{b_{22}}{a_{22}}\right)\cdot\frac{\partial}{\partial\delta_{ij}}\left(\frac{b_{22}}{a_{22}}\right)\right\} \qquad (6\text{-}19)$$

The stiffness matrix KNB(48,48) is

$$\frac{\partial^2 Q_1}{\partial\delta_{ij}\partial\delta_{nm}} = \beta_1\left\{\left(\frac{b_{11}}{a_{11}}\right)\cdot\frac{\partial^2}{\partial\delta_{ij}\partial\delta_{nm}}\left(\frac{b_{11}}{a_{11}}\right)+\frac{\partial}{\partial\delta_{nm}}\left(\frac{b_{11}}{a_{11}}\right)\cdot\frac{\partial}{\partial\delta_{ij}}\left(\frac{b_{11}}{a_{11}}\right)\right.$$

$$\left.\left(\frac{b_{22}}{a_{22}}\right)\cdot\frac{\partial^2}{\partial\delta_{ij}\partial\delta_{nm}}\left(\frac{b_{22}}{a_{22}}\right)+\frac{\partial}{\partial\delta_{nm}}\left(\frac{b_{22}}{a_{22}}\right)\cdot\frac{\partial}{\partial\delta_{ij}}\left(\frac{b_{22}}{a_{22}}\right)\right\} \qquad (6\text{-}20)$$

To simplify the solution, the double integration can be carried out after working out the differentiation as in equations (6-19) and (6-20)

6-2-2 Calculation of the geodesic bending term (Q_2)

The second term in equation (6-16) Q_2, can be worked out by using the tensor analysis and the surface fundamental forms as follows;

$$Q_2 = \frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\frac{a_{1,1}\cdot\left(a_3 X a_1\right)}{\sqrt{\left(a_{11}\right)^3}}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\frac{a_{2,2}\cdot\left(a_3 X a_2\right)}{\sqrt{\left(a_{22}\right)^3}}\right\}^2 d\theta_1 d\theta_2 \qquad (6\text{-}21)$$

According to tensor notation the vector multiplication in the above equation can be treated as shown below;

$$a_3 \times a_1 = \epsilon_{12} a^2$$

and $$a_3 \times a_2 = \epsilon_{21} a^1 \qquad (6\text{-}22)$$

where $\epsilon_{12} = -\epsilon_{21} = \sqrt{a}$

$\epsilon_{12}, \epsilon_{21}$ are special skew symmetric tensors.

By substituting in equation (6-21) then,

$$Q_2=\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\frac{\sqrt{a}\left(a_{1,1}.a^2\right)}{\sqrt{\left(a_{11}\right)^3}}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\frac{\sqrt{a}\left(a_{2,2}.a^1\right)}{\sqrt{\left(a_{22}\right)^3}}\right\}^2 d\theta_1 d\theta_2 \qquad (6\text{-}23)$$

The Christoffel symbols of the second kind with respect to the surface can be found from the scalar product as,

$$\Gamma^2_{11}= a_{1,1}.a^2 \quad \text{and} \quad \Gamma^1_{22}= a_{2,2}.a^1 \qquad (6\text{-}24)$$

then the equation (6-23) may be rewritten in form,

$$Q_2=\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\frac{\sqrt{a}\left(\Gamma^2_{11}\right)}{\sqrt{\left(a_{11}\right)^3}}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\beta_2\left\{\frac{\sqrt{a}\left(\Gamma^1_{22}\right)}{\sqrt{\left(a_{22}\right)^3}}\right\}^2 d\theta_1 d\theta_2 \qquad (6\text{-}25)$$

Then the value of the equivalent load on the control point and the stiffness term of the geodesic bending can be calculated from the first and second derivative of equation (6-25);

The equivalent load on the control point FGB= $\frac{\partial Q_2}{\partial \delta_{i\,j}}$ where;

$$\frac{\partial Q_2}{\partial \delta_{ij}} = \beta_2 \left\{ \left[\frac{\sqrt{a}\left(\Gamma^2_{11}\right)}{\sqrt{\left(a_{11}\right)^3}} \right] \cdot \frac{\partial}{\partial \delta_{ij}} \left[\frac{\sqrt{a}\left(\Gamma^2_{11}\right)}{\sqrt{\left(a_{11}\right)^3}} \right] + \left[\frac{\sqrt{a}\left(\Gamma^1_{22}\right)}{\sqrt{\left(a_{22}\right)^3}} \right] \cdot \frac{\partial}{\partial \delta_{ij}} \left[\frac{\sqrt{a}\left(\Gamma^1_{22}\right)}{\sqrt{\left(a_{22}\right)^3}} \right] \right\} \qquad (6\text{-}26)$$

The stiffness matrix KGB(48,48) is,

$$\frac{\partial^2 Q_2}{\partial \delta_{ij} \partial \delta_{nm}} = \beta_2 \left\{ \left[\frac{\sqrt{a}\left(\Gamma^2_{11}\right)}{\sqrt{\left(a_{11}\right)^3}} \right] \cdot \frac{\partial^2}{\partial \delta_{ij} \partial \delta_{nm}} \left[\frac{\sqrt{a}\left(\Gamma^2_{11}\right)}{\sqrt{\left(a_{11}\right)^3}} \right] + \frac{\partial}{\partial \delta_{nm}} \left[\frac{\sqrt{a}\left(\Gamma^2_{11}\right)}{\sqrt{\left(a_{11}\right)^3}} \right] \cdot \frac{\partial}{\partial \delta_{ij}} \left[\frac{\sqrt{a}\left(\Gamma^2_{11}\right)}{\sqrt{\left(a_{11}\right)^3}} \right] \right.$$

$$\left. + \left[\frac{\sqrt{a}\left(\Gamma^1_{22}\right)}{\sqrt{\left(a_{22}\right)^3}} \right] \cdot \frac{\partial^2}{\partial \delta_{ij} \partial \delta_{nm}} \left[\frac{\sqrt{a}\left(\Gamma^1_{22}\right)}{\sqrt{\left(a_{22}\right)^3}} \right] + \frac{\partial}{\partial \delta_{nm}} \left[\frac{\sqrt{a}\left(\Gamma^1_{22}\right)}{\sqrt{\left(a_{22}\right)^3}} \right] \cdot \frac{\partial}{\partial \delta_{ij}} \left[\frac{\sqrt{a}\left(\Gamma^1_{22}\right)}{\sqrt{\left(a_{22}\right)^3}} \right] \right\} \qquad (6\text{-}27)$$

The double integration can be carried out after the differentiation of the last two equations.

6-2-3 Calculation of the torsion bending term (Q_3)

The torsion bending term can be written by using the tensor notation as follows,

$$Q_3 = \frac{1}{2} \int_{\theta_1=0}^{1} \int_{\theta_2=0}^{1} \gamma \cdot \left\{ - \frac{\left(a_3 X a_1\right) \cdot a_{3,1}}{\left(a_{11}\right)} \right\}^2 d\theta_1 d\theta_2$$

$$+ \frac{1}{2} \int_{\theta_1=0}^{1} \int_{\theta_2=0}^{1} \gamma \cdot \left\{ - \frac{\left(a_3 X a_2\right) \cdot a_{3,2}}{\left(a_{22}\right)} \right\}^2 d\theta_1 d\theta_2 \qquad (6\text{-}28)$$

By using the relation of the second fundamental form of the surface in equation (6-28), therefore;

$$Q_3=\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\gamma.\left\{-\frac{\frac{1}{\sqrt{a}}\left(a_{12}b_{11}-a_{11}b_{12}\right)}{\left(a_{11}\right)}\right\}^2 d\theta_1 d\theta_2$$

$$+\frac{1}{2}\int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1}\gamma.\left\{-\frac{\frac{1}{\sqrt{a}}\left(a_{22}b_{21}-a_{12}b_{22}\right)}{\left(a_{22}\right)}\right\}^2 d\theta_1 d\theta_2 \qquad (6\text{-}29)$$

Then the value of the equivalent load of the torsion bending on the control point can be calculated from the first differentiation of equation (6-28);

The equivalent load on the control point FGB= $\frac{\partial Q_3}{\partial \delta_{ij}}$

where;

$$\frac{\partial Q_3}{\partial \delta_{ij}}=\gamma\left\{\left[\frac{\frac{1}{\sqrt{a}}\left(a_{12}b_{11}-a_{11}b_{12}\right)}{\left(a_{11}\right)}\right].\frac{\partial}{\partial \delta_{ij}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{12}b_{11}-a_{11}b_{12}\right)}{\left(a_{11}\right)}\right]\right.$$

$$\left.+\left[\frac{\frac{1}{\sqrt{a}}\left(a_{22}b_{21}-a_{12}b_{22}\right)}{\left(a_{22}\right)}\right].\frac{\partial}{\partial \delta_{ij}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{22}b_{21}-a_{12}b_{22}\right)}{\left(a_{22}\right)}\right]\right\} \qquad (6\text{-}30)$$

From the second derivative of equation (6-28), the stiffness matrix of the torsion bending can take the form,

$$KBT(48,48)=\frac{\partial^2 Q_3}{\partial \delta_{ij}\,\partial \delta_{nm}}$$

where;

$$\frac{\partial^2 Q_3}{\partial\delta_{ij}\partial\delta_{nm}} = \gamma\left\{\left[\frac{\frac{1}{\sqrt{a}}\left(a_{12}b_{11} - a_{11}b_{12}\right)}{\left(a_{11}\right)}\right]\cdot\frac{\partial^2}{\partial\delta_{ij}\partial\delta_{nm}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{12}b_{11} - a_{11}b_{12}\right)}{\left(a_{11}\right)}\right]\right.$$

$$+\frac{\partial}{\partial\delta_{nm}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{12}b_{11} - a_{11}b_{12}\right)}{\left(a_{11}\right)}\right]\cdot\frac{\partial}{\partial\delta_{ij}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{12}b_{11} - a_{11}b_{12}\right)}{\left(a_{11}\right)}\right]$$

$$+\left[\frac{\frac{1}{\sqrt{a}}\left(a_{22}b_{21} - a_{12}b_{22}\right)}{\left(a_{22}\right)}\right]\cdot\frac{\partial^2}{\partial\delta_{ij}\partial\delta_{nm}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{22}b_{21} - a_{12}b_{22}\right)}{\left(a_{22}\right)}\right]$$

$$\left.+\frac{\partial}{\partial\delta_{nm}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{22}b_{21} - a_{12}b_{22}\right)}{\left(a_{22}\right)}\right]\cdot\frac{\partial}{\partial\delta_{ij}}\left[\frac{\frac{1}{\sqrt{a}}\left(a_{22}b_{21} - a_{12}b_{22}\right)}{\left(a_{22}\right)}\right]\right\} \qquad (6\text{-}31)$$

The double integration can be carried out after the differentiation of equations (6-30).(6-31)

6-2-4 Calculation of equivalent load and stiffness matrices of the spring supports term (Q_4)

In the three dimensional analysis,three translations and three rotations must be constrained to keep any structure in a stable mode. In this analysis, three springs will be used at certain nodes along the boundary to prevent any translation. Therefore, the stiffness of the springs and its position must be defined in the data.

The strain energy equation of the spring can be written as,

$$Q_4 = \frac{1}{2} \quad S\delta_1^2 \tag{6-32}$$

the equivalent load and the stiffness terms, can be calculated from the first and the second derivatives of the above expression.

$$P_{ij} = \frac{\partial Q_4}{\partial X_{ij}} = S.\delta \frac{\partial \delta}{\partial X_{ij}} \tag{6-33}$$

$$KS(48,48)= S. \frac{\partial \delta}{\partial X_{ij}} . \frac{\partial \delta}{\partial X_{mn}} + S.\delta \frac{\partial^2 \delta}{\partial X_{ij} \partial X_{mn}} \tag{6-34}$$

To simplify the solution of the equation.(6-34) the second part of that equation can be neglected due to smallness, so that the equation can be written in this formula;

$$KS(48,48)= S. \frac{\partial \delta}{\partial X_{ij}} \frac{\partial \delta}{\partial X_{mn}} \tag{6-35}$$

where, δ is = (X_{ij} component of the **r**) - constant.

This constant is the initial value of the spring position defined in the data as has been mentioned in chapter 5.

then, equations (6-33) and (6-34) can put in the following form;

$$P(48)=S.\left[\left(D(i,j).X_{ij} - \text{constant}\right).D(i,j)\right] \tag{6-36}$$

$$KS(48,48)=S.D(i,j).D(m,n) \tag{6-37}$$

where; the values of D(i,j) has been defined in chapter five.

6-2-5 Calculation of the potential energy of the external load (Q_5)

The loads which are applied to a surface must be supplied in the data which usually start with an arbitrary load and increase by a small increment. For many problems the majority of the nodes do not have any loads applied to them, so that the loaded nodes need only to be defined in the data.

The potential energy of the applied load is $(U)=-P.\Delta$ so, the equivalent load on the control points are;

$$\frac{\partial \Delta}{\partial X_{ij}} = \Sigma \frac{\partial U}{\partial X_{ij}} \qquad (6\text{-}38)$$

where ij = no. of degree of freedom

then, the equivalent force on the control points (FP) may be in the form;

$$FP(48) = P.\frac{\partial \Delta}{\partial X_{ij}} = P.D(i,j) \qquad (6\text{-}39)$$

6-4 Membrane forces in the grid shell

An iteration method is carried out to updating the coordinates of control points until the strain reach to or close to zero. The change of strain in each element of the surface can be calculated as follows,

- each element may be divided into a number of small elements in

both directions u and v.

- the elongation in the two direction can be calculated by assuming an initial value of a_{11} and a_{22} ,then the strain can be defined for the small element in u and v directions.
- by integrating those values, the strain (ε) and the stiffness value of R and Q in the matrix (6-5) can be calculated as follows;

$$\varepsilon_1 = \frac{a_{11} - A_{11}}{\sqrt{(A_{11})}} \tag{6-40}$$

$$\varepsilon_2 = \frac{a_{22} - A_{22}}{\sqrt{(A_{22})}} \tag{6-41}$$

where, A_{11},A_{22} are the initial value of a_{11},a_{22} respectively

by using the first derivative of the above equation, the value of $\left\{\frac{\partial R_{11}}{\partial \delta_{11}}\right\}$ and $\left\{\frac{\partial R_{22}}{\partial \delta_{22}}\right\}$ can be defined in the form,

$$\left\{\frac{\partial R_{11}}{\partial \delta_{11}}\right\} = \left\{\frac{\partial Q_{11}}{\partial \delta_{11}}\right\}^T = -\frac{\frac{\partial a_{11}}{\partial u}}{2A_{11}} \tag{6-42}$$

$$\left\{\frac{\partial R_{22}}{\partial \delta_{22}}\right\} = \left\{\frac{\partial Q_{22}}{\partial \delta_{22}}\right\}^T = -\frac{\frac{\partial a_{22}}{\partial u}}{2A_{22}} \tag{6-43}$$

6-5 Results and Conclusions

The grid shell which has been analyzed is composed of four elements as shown in Fig.(6-1). The shape of the shell is defined by;

$$Z = A \cosh \frac{X}{A} + B \cosh \frac{Y}{B} \qquad (6\text{-}44)$$

where,

X,Y and Z are the coordinates of the control points.

A and B are constants.

The behaviour of the surface is controlled by the 25 control points shown in Fig.(6-2). Also this figure shows the marked points where the springs have been used in three directions.

Two cases of loading were studied, the first case was symmetric in which the load was applied at the centre of the shell, and the second was asymmetric where the load was applied at the middle of the first element.

From the previous analysis of the two cases the following results have been achieved;

1- The grid shells in this example behaves purely elastic in both cases.

2- No difference in results occurs when the element is divided into nine small elements (3x3) and when it is divided into four element (2x2) during the calculation of the integration of the bending and torsion.

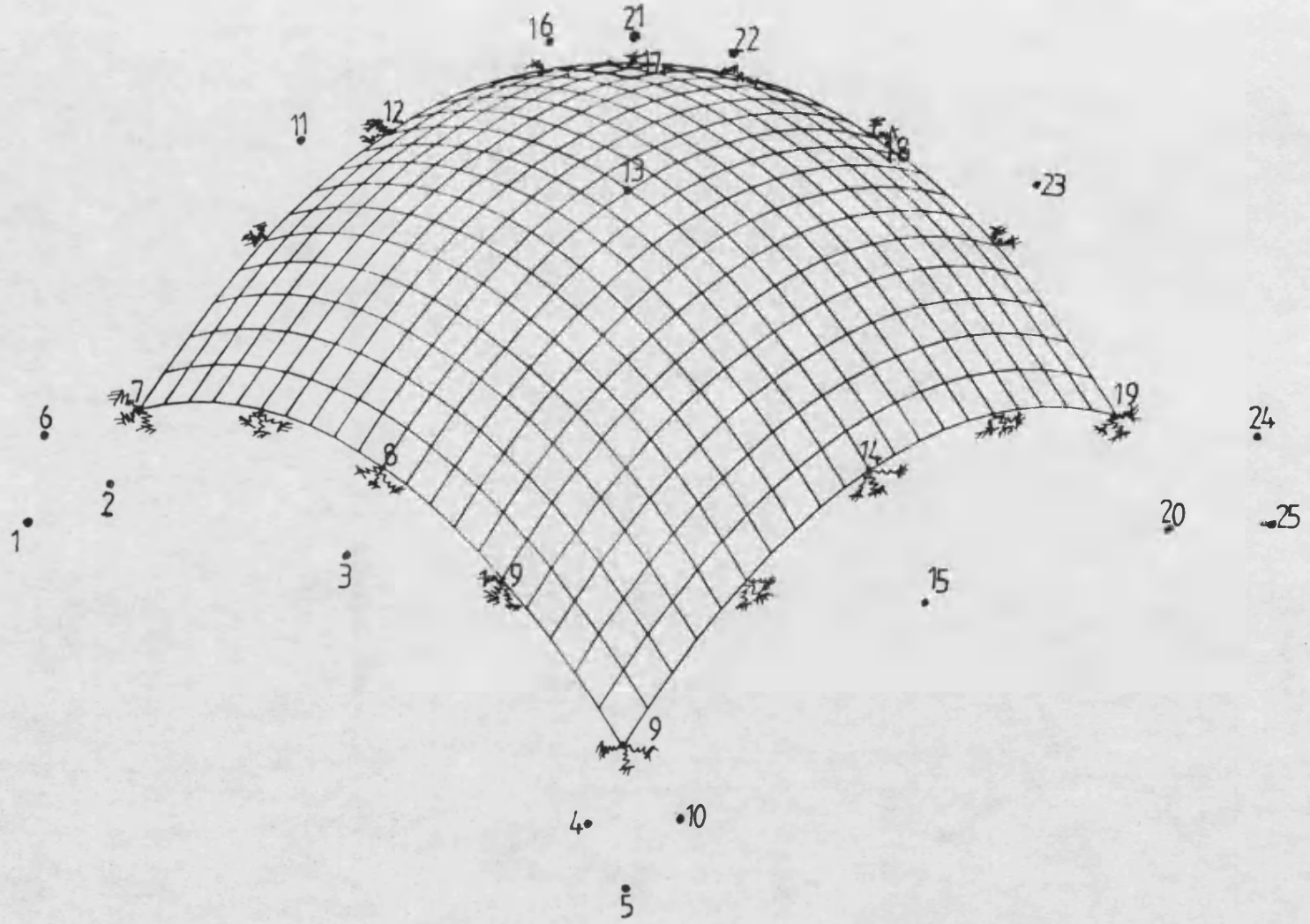


Fig.(6-2) The control points and springs positions on the grid shell

3- In the strain calculations, each element is divided in four small elements (2x2), where eight values of strain for each element (2 direction x 4 small elements) have been obtained. That means, the number of constrains were 32 against 75 D.O.F (25 control points x 3 directions).

4- The program did not converge when the element was divided into nine elements (3x3), even when the element was taken as a whole.

From the above remarks, the following conclusions can be written;

1- The division of the elements during integration did not affect the values of the bending and torsion stiffness and forces.

2- Care must be taken during the division of the elements of the when calculating the strain.

3- The ratio between the number of constraints and the degrees of freedom of the control points may be in the range of 1:2

4- In the author's opinion the previous point has no right explanation and thus, more application or further research may be needed

From the above analysis a computer program has been made ,(it will be discussed in Appendix C-2). Fig.(6-3) and Fig.(6-4) shows the behaviour of two applications that have been loaded symmetirc-ally and unsymmetrically respectively.

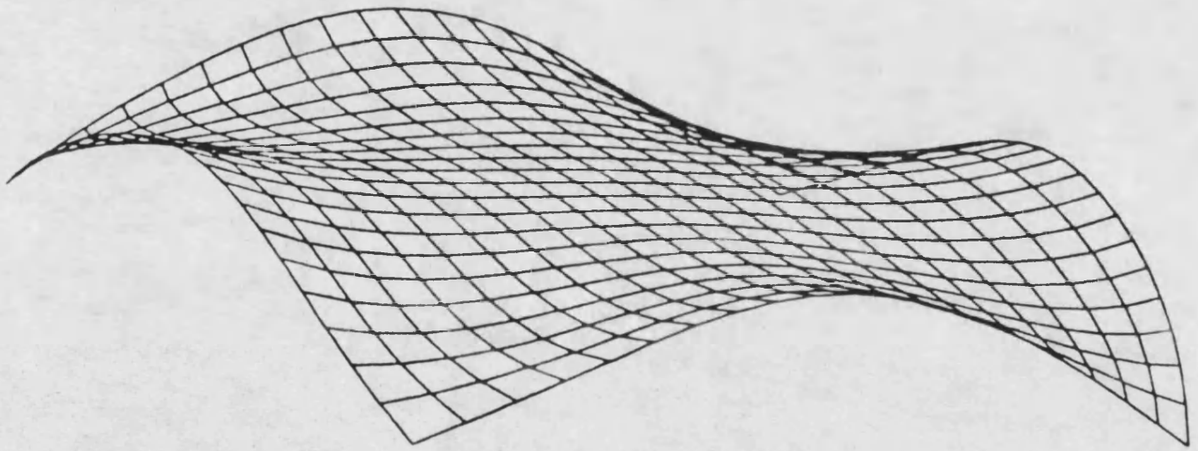


Fig.(6-3-a) Symmetric load (deformation at 20.0 kg.)

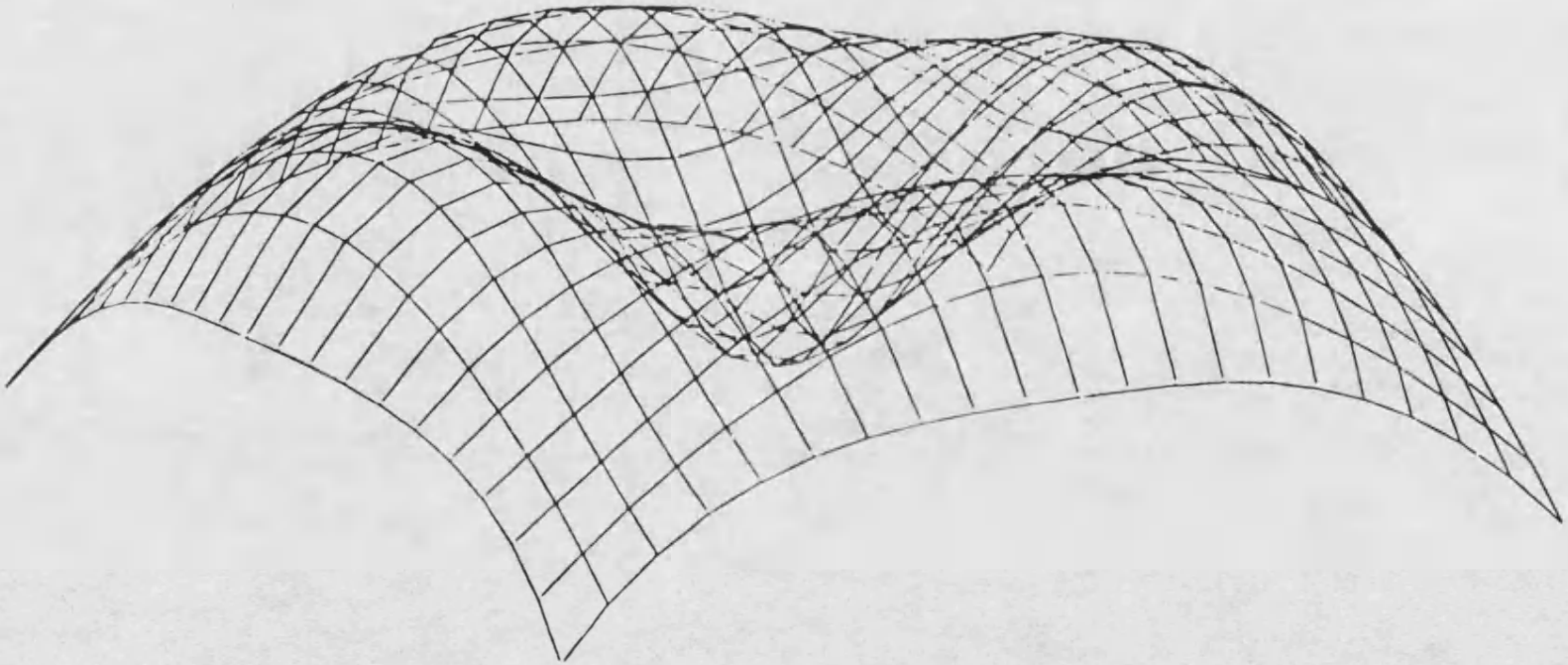


Fig.(6-3-b) Symmetric load (deformation at 90.0 kg.)

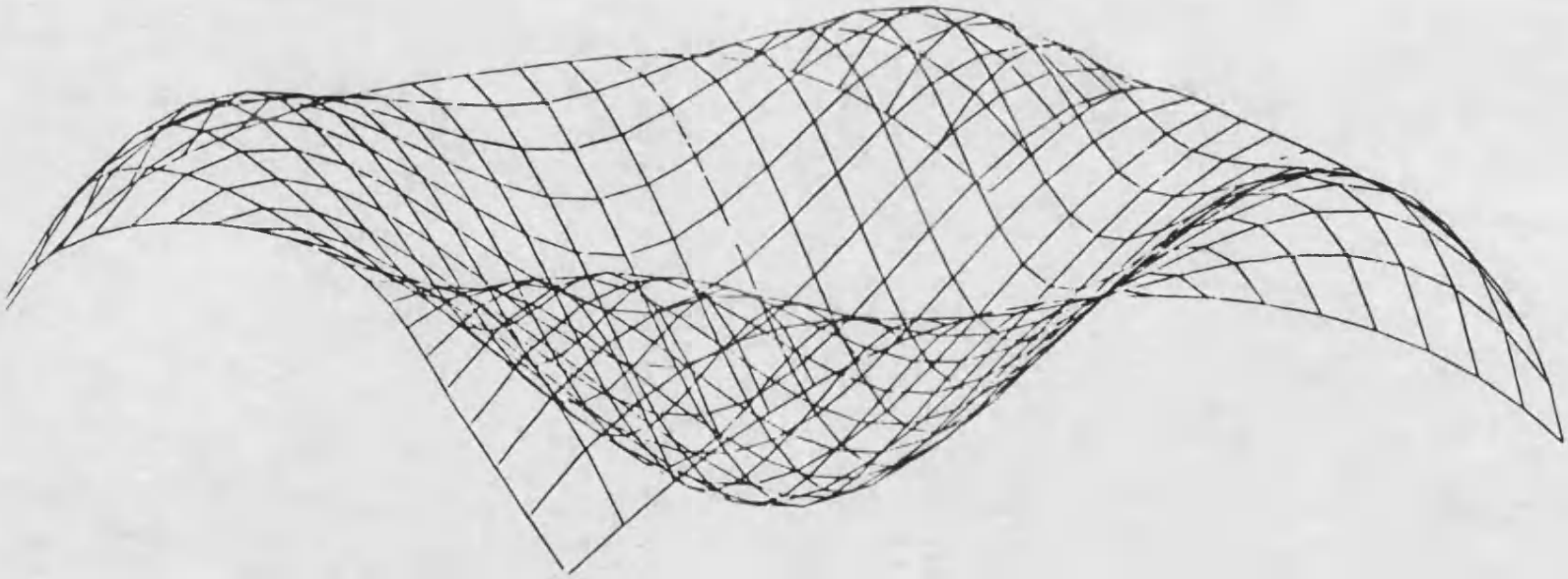


Fig.(6-3-c) Symmetric load (deformation at 150.0 kg.)

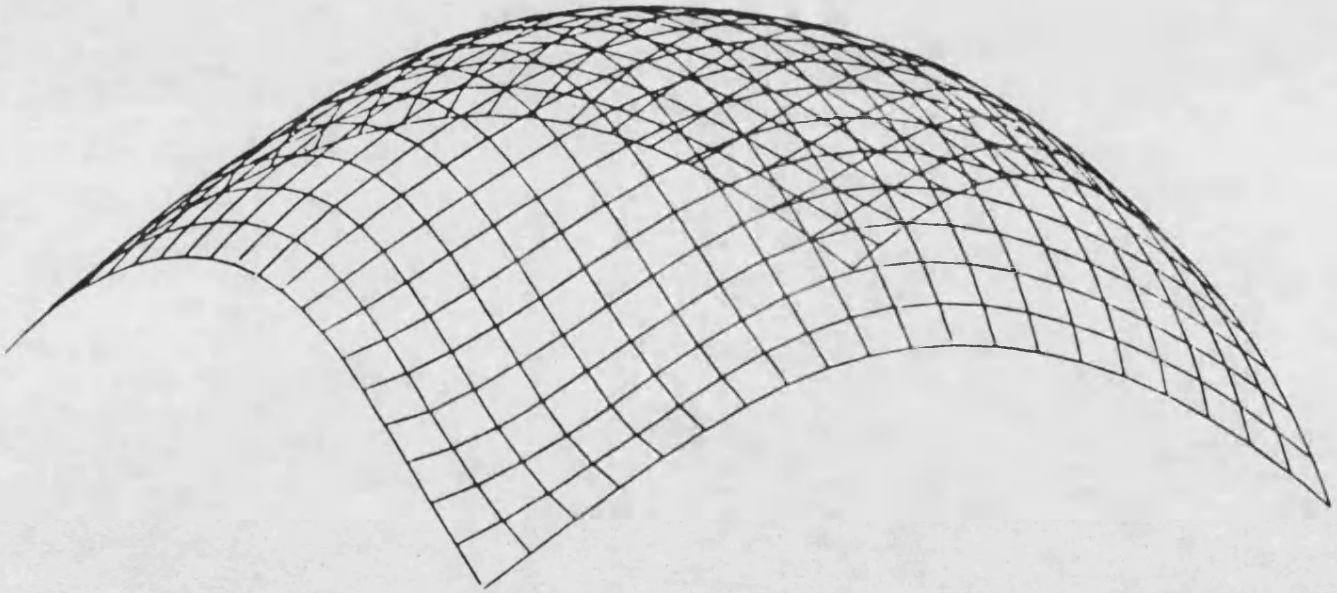


Fig.(6-4-a) Unsymmetrical loading (deformation at 10.0 kg.)

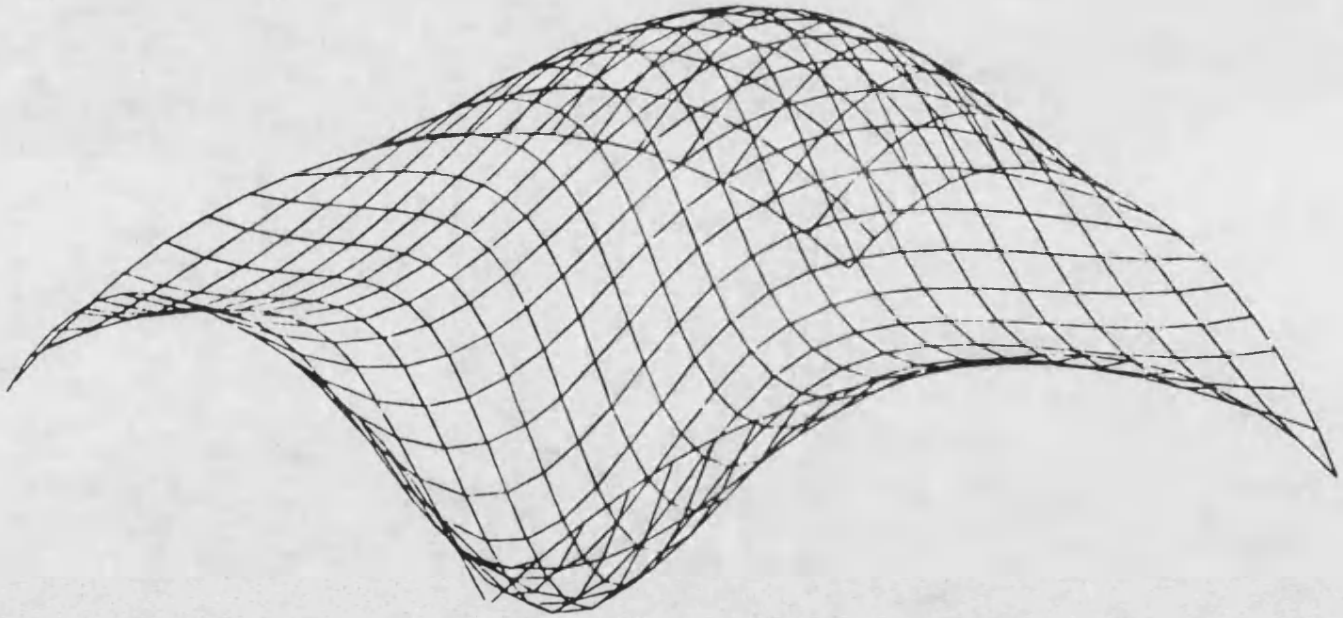


Fig.(6-4-b) Unsymmetrical loading (deformation at 30.0 kg.)

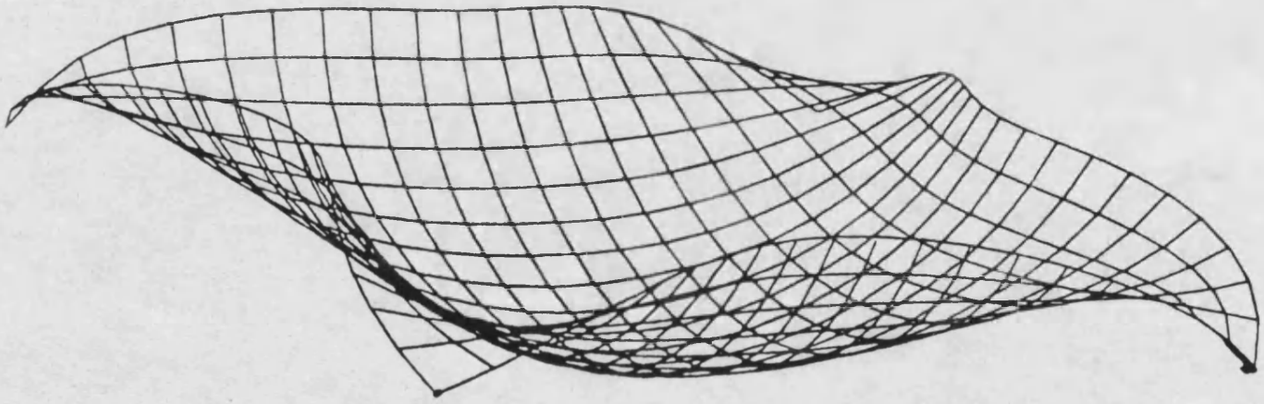


Fig.(6-4-c) Unsymmetrical loading (deformation at 45.0 kg.)

CHAPTER SEVEN

EXPERIMENTAL STUDY OF THE BEHAVIOUR OF GRID SHELLS

7-1 Introduction

Most of the studies made on the construction of grid shells were combined with experimental investigations on physical models.

This chapter describes an experimental study to investigate the behaviour of a model of a grid shell under different concentrated loads. The model test and its method of construction are discussed and the apparatus used in the test is described. The experimental results are presented at the end of this chapter and also compared with the corresponding behaviour predictions, obtained using the computer program of the analysis of grid shells.

In this test two models are used, the first one studies the behaviour of grid shells under unsymmetrical load, where as the second model is under full symmetric loading.

The mathematical calculation of the model test is shown in Appendix D.

7-2 Description of model structure

7-2-1 Configuration

The two models tested were single layer domes fixed to the edges of a plywood frame built especially for this test, Fig.(7-1) shows the wooden frame. The surface was made from self colour mild steel woven wire mesh 2x2, and the space between the wires was 1.28cm. By using the woven wire mesh, joints between the crossing wires, were free in rotation. The configurations of the two surfaces were calculated from the formula of equation (6-31) in which;

$$Z = A \cosh \frac{X}{A} + B \cosh \frac{Y}{B}$$

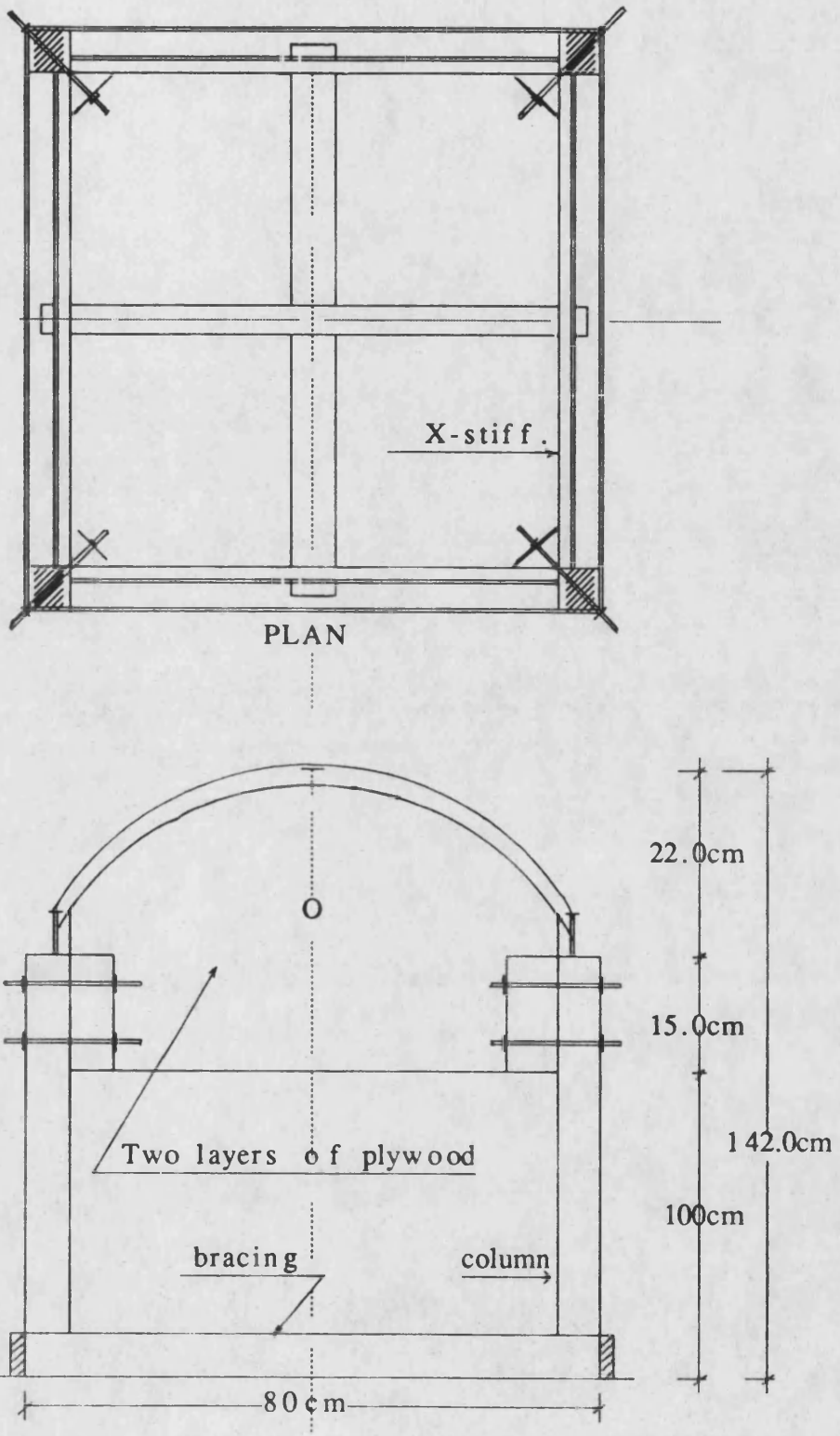
where;

A and B are constant, (equal to 50)

X,Y and Z are the dimensions of the dome nodes.

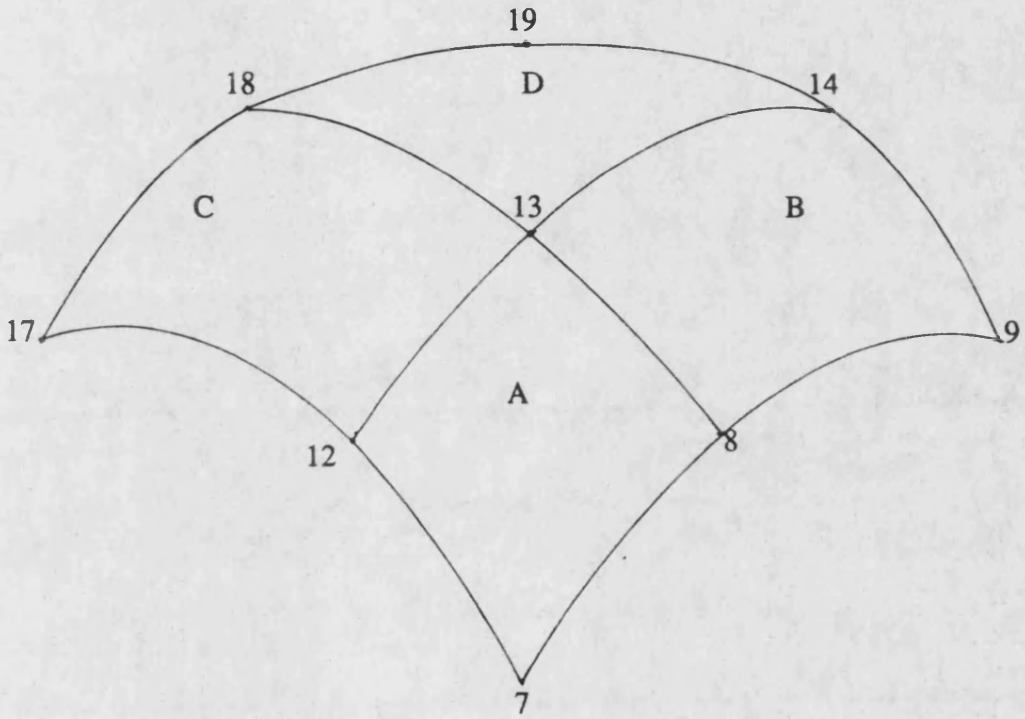
The surface was divided into 4 patches with 9 nodes, as shown in Fig.(7-2). Theoretically each patch was controlled by 16 control points and the overlapping between them produced 25 control points for all the surface, the geometry of the control points of the model are shown in Table (7-1).

The supports of the dome which are considered as semi-rigid, are continuous along the edges of the frame to prevent any translation.



Elevation or side-view

Fig.(7-1) The testing model



The surface model(4 patches)

Fig(7-2)

Node No.	X(cm.)	Y(cm.)	Z(cm.)
1	-80.00	-80.00	00.00
2	-40.00	-80.00	62.00
3	00.00	-80.00	78.87
4	40.00	-80.00	62.00
5	80.00	-80.00	00.00
6	-80.00	-40.00	62.00
7	-40.00	-40.00	124.00
8	00.00	-40.00	140.90
9	40.00	-40.00	124.00
10	80.00	-40.00	62.00
11	-80.00	00.00	78.87
12	-40.00	00.00	140.90
13	00.00	00.00	157.74
14	40.00	00.00	140.90
15	80.00	00.00	78.87
16	-80.00	40.00	62.00
17	-40.00	40.00	124.00
18	00.00	40.00	140.90
19	40.00	40.00	124.00
20	80.00	40.00	62.00
21	-80.00	80.00	00.00
22	-40.00	80.00	62.00
23	00.00	80.00	78.87
24	40.00	80.00	62.00
25	80.00	80.00	00.00

Table(7-1) theoretical geometry of the control points

7-2-2 Constructional details of the model

The construction of the testing model was divided into two parts. The first part was concerned with the design of the wooden frame, taking into consideration the requirement to use this frame throughout all the testing without damage. The second part was concerned with the wire mesh (i.e. grid shells) which needed to be changed after every test. The shape of the frame as shown in Fig.(7-1) was constructed to fulfill the following conditions.

a- Provide the exact shape of the wire mesh.
b- To be stiff enough to prevent deformation during the test loading.
c- To be easy to load the grid shell and monitor the deformation.

Therefore, the construction of the wooden frame passed through the following steps;

1- Each of the four sides consists of two layers of plywood thickness 12.0 mm, having the same curvature with 5.0 cm difference in height between them. This difference provides a groove for the wire to rest on. A polyester resin was used to fix the two layers together and four bolts with nuts and washers were also used to prevent any separation between the plywood sheets during testing, see Fig.(7-3-a).
2- Every two sides were jointed to a wooden corner to adjust the level, then a steel angle was used to fully tighten the corner

to prevent any collapse in the frame corner's, see Fig(7-3-b).

3- For every test, the edges of the frame were covered by a plastic tape before fixing the wire to keep the frame undamaged for the next test (i.e. acting as a release agent).

4- Two wooden rods ϕ 25.0 mm were used as ties across opposing sides to stiffen the frame across

5- Four wooden posts of height 1.0 m and X-section 100x100 mm^2 were connected to the frame corner's using two threaded steel rods 250 mm in length with nuts for each post. By lifting the level of the model by 1.0 m, the fixation of the surface and the loading process would be more easier. Horizontal and cross bracing were used between the posts to prevent any movement of the model, see Fig.(7-3-c).

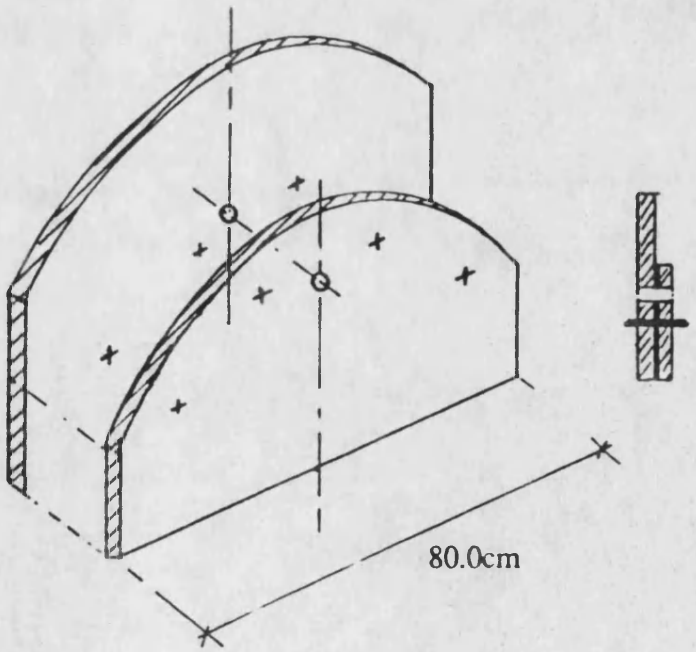


Fig.(7-3-a)

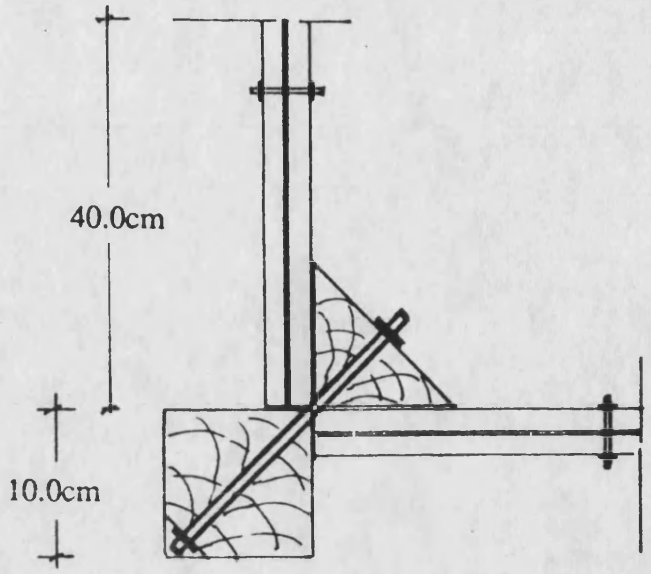


Fig.(7-3-b)

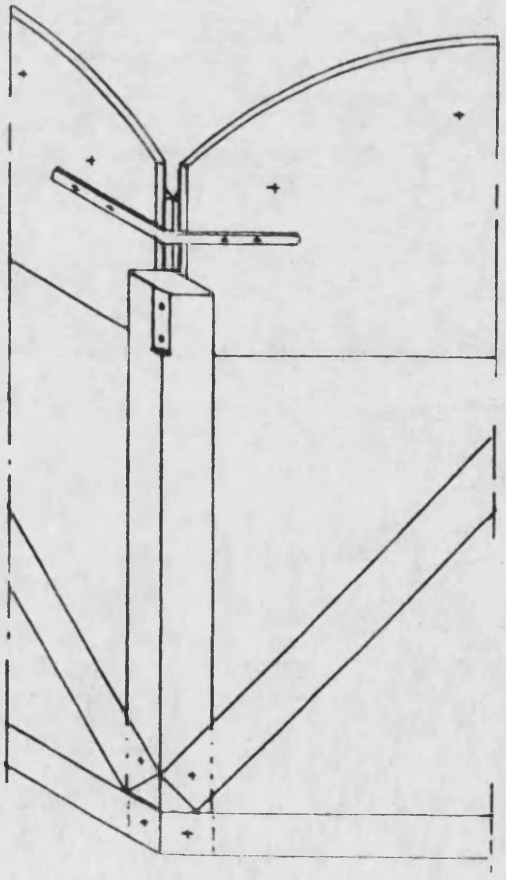


Fig.(7-3-c)

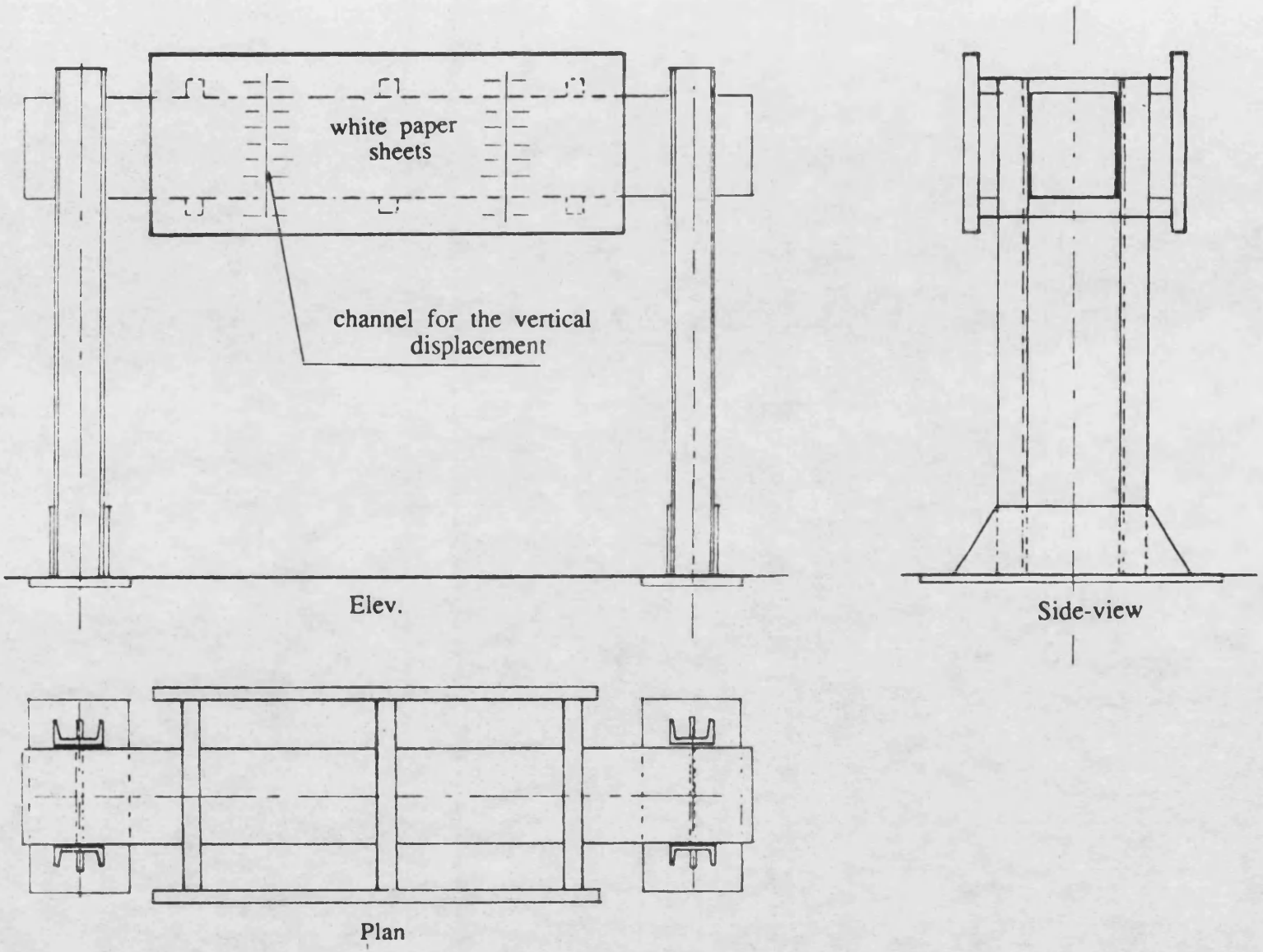
According to the calculation of the model dimensions, the required area of the wire mesh was cut and the sharp ends bent. By pushing the mesh from its four sides at the same time, fitting it to the frame, the model configuration was obtained. The primary fixation for surface was made using staples along the model's edge at spacings of about 150 mm . Also, the staples were able to hold the surface if it was subjected to tension forces during the loading. Finally the full joining between the surface and the frame was achieved by using polyester resin (IPSON P.38) along all the edges. This type of fixation needed at least 24 hours in order to test it under full strength.

7-3 The testing rig

The function of the testing rig, shown in Fig.(7-4), was to provide a strong frame clamp to the wooden model to ensure that there were no displacements or translation during the loading. Also to facilitate the loading and the monitoring of the resulting deformations during the experiments performed on the grid shell.

7-3-1 Description of the testing rig

The rig consisted of two steel columns and a wooden beam clamped to them. The base of the columns were connected to the lab floor by bolts. Six transverse wooden beams (380 mm in length and 50x50 mm^2 in cross section) were connected to the main beam. Then two flat plywood sheets were fixed on the two sides and covered by



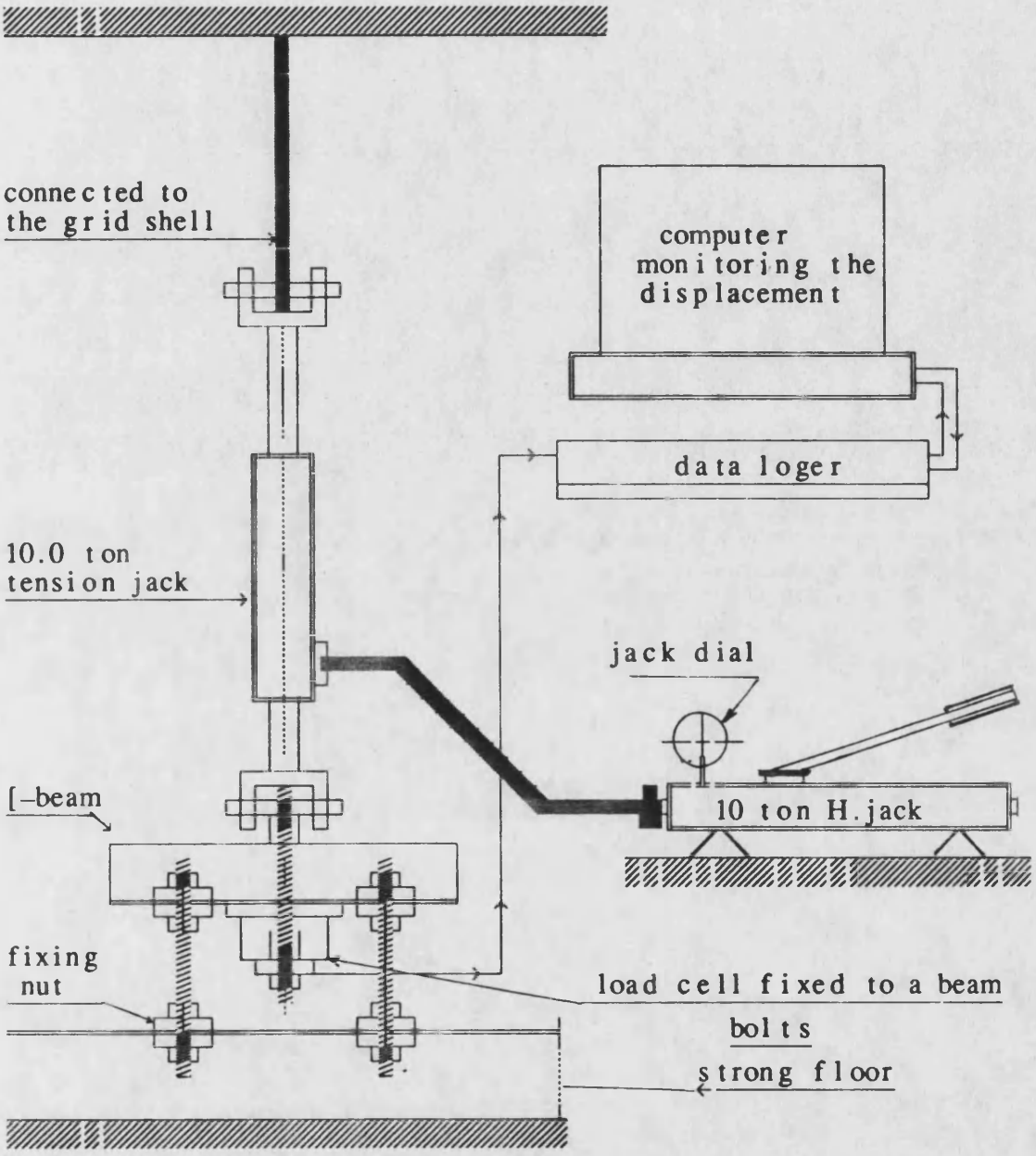
Fig(7-4) The testing rig

white paper sheets see Fig.(7-4).

7-3-2 Loading system

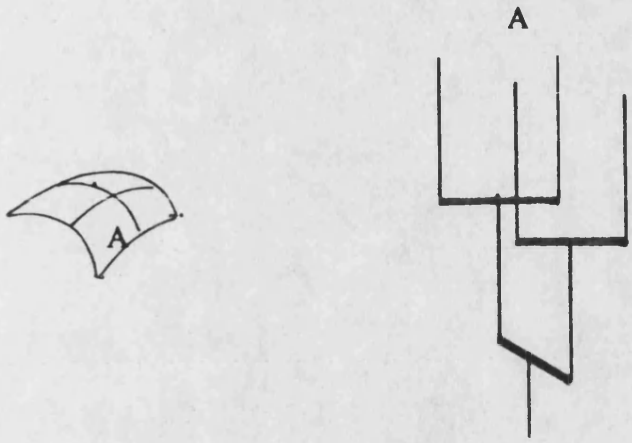
The loading of the models was achieved by loading the appropriate nodes of the dome using a system of steel cables and spreader beams (wiffle tree) connected to a 10.0 ton tension jack fixed to the strong floor of the lab . A load cell was placed between the screw jack and the loading arrangement, where the load cell was connected to a datalogar and a computer. Therefore, the load on the surface was recorded by the load cell through the computer (see Fig.(7-5).

Two cases of loading were considered. In the first case of loading, the load was applied at four nodes of one corner to produce an unsymmetrical load as shown in Fig.(7-6-a). In the second case full symmetric loads were applied on sixteen points ,four points at each quarter as shown in Fig. (7-6-b).



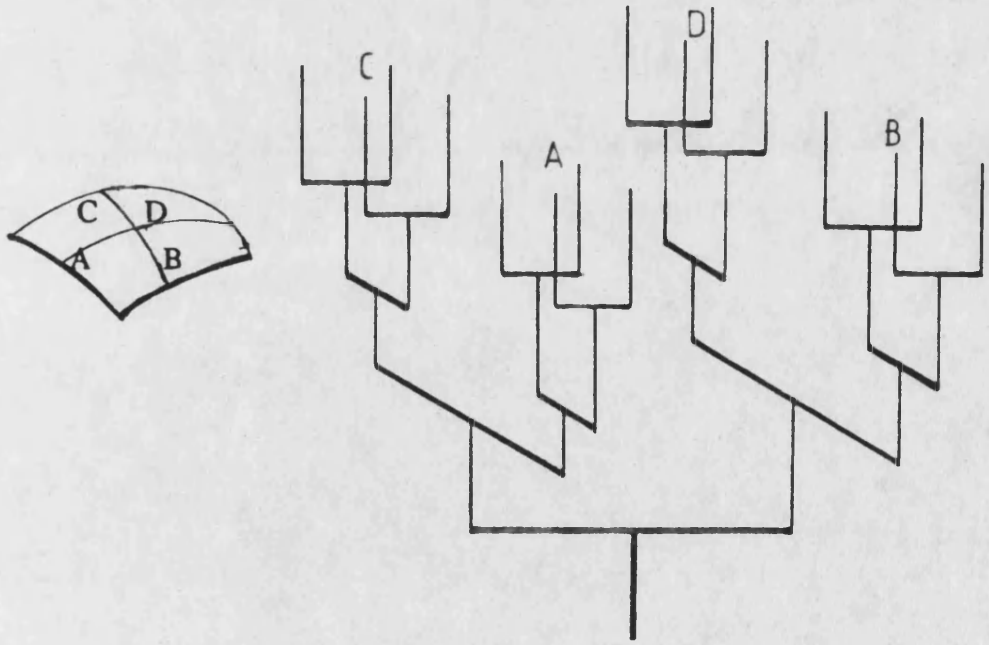
The loading system

Fig.(7-5)



Case of loading No.1

Fig.(7-6-a)



case of loading No. 2

Fig.(7-6-b)

7-3-3 Deformation measurements

Vertical displacements were recorded at four nodes (at the centre of each quarter). A steel rod, 1.0 mm diameter, soldered vertically to those nodes , and at each position a channel was made on the wooden beam of the testing rig to control the node displacement vertically. A scale was marked at the position of the rods, see Fig.(7-4).

7-4 Determination of elastic properties of the wire mesh

It is necessary during the comparison between the theoretical and experimental studies, to determine the elastic properties of the wire mesh used in the model to interpret the deflections and to predict the collapse load. Therefore two tests were carried out to measure the bending stiffness of the wire mesh, flexural and tension test.

7-4-1 Flexural test

Two types of flexural tests were carried out on a sample of the wire used in the mesh (five samples were used in each test). The first test treated the sample as a simply supported beam (S.S.B), set up between two roller supports . Load was applied via a knife edge to the centre of the test specimen and the vertical deflection of the central points were recorded for a given applied load, see Fig.(7-7-a). The second test treated the

sample as a cantilever fixed at one end and the load was applied on the other, also the applied load and the corresponding deflection at the end were recorded, see Fig.(7-7-b). The results of the two tests are shown in table (7-2) .

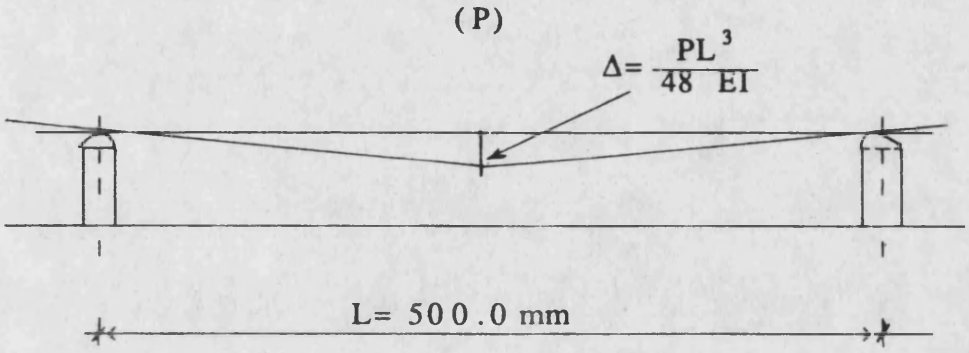


Fig. (7 - 7 - a)

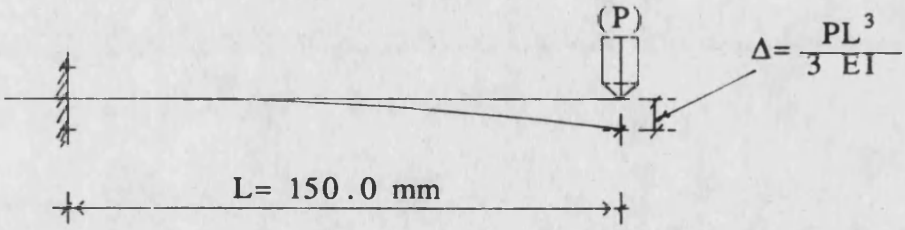


Fig.(7-7-b) Arrangement of sample for flexural tests.

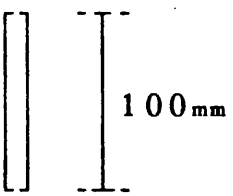
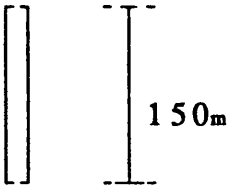
TEST TYPE	SAMPLE NO.	LOAD (gm)	DEFLEC. (mm)	(EI) VALUES	SPECIMEN DIMEN.
(1) S.S.B. $\Delta = \frac{PL(A.LE)^2}{48\ EI}$	1	100	47	57.65	 100mm
	2		48	56.44	
	3		50	54.19	
	4		49	55.29	
	5		60	45.15	
(2) CANT. $\Delta = \frac{PL^3}{3\ EI}$	1	90	20	50.82	 150m
	2		25	40.50	
	3		22	46.50	
	4		24	42.18	
	5		20	50.82	
Average EI	49.867				

Table (7-2) Results of flexural tests

7-4-2 Tensile test

The tensile test was carried out on five specimens of the steel wire used in the mesh. The samples were 300 mm in length. The diameter of the specimens were determined using a micrometer (where ϕ 1.63 mm). The 10.0 ton DARTEC test machine was used to measure the applied loads and the corresponding elongation, see Fig.(7-8). From the machine plots, the load- displacement relation was plotted for each specimen and the value of

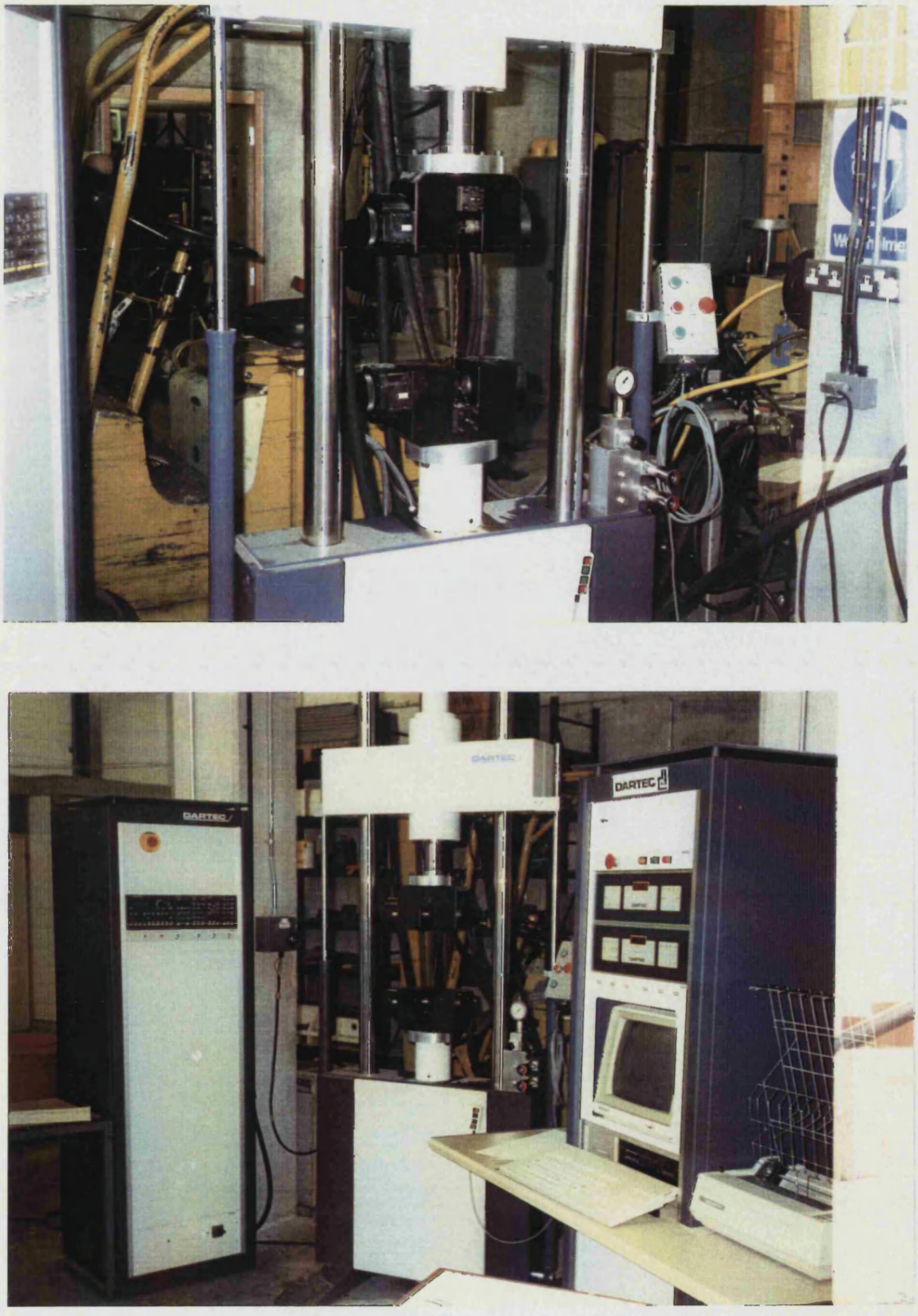


Fig.(7-8) The tension test using the 10 t DARTEC machine

the modulus of elasticity (E) was calculated. Figures (7-9-a) and (7-9-b) shows the relation between the load and the corresponding deflection, (Fig.(7-9-b) amplified the first 20.0 mm of displacement of Fig.(7-9-a)).

7-5 First case of loading

One quarter only of the surface was loaded to achieve the unsymmetrical case, the wiffle tree was hanged at four points in element no. 1, see Fig. (7-5-a). The starting load was 5.0 kg and increased by 5.0 kg until the surface collapsed. The relation between the applied load and the corresponding deflection was recorded at the middle of each quarter. During the early loading stage, the loaded quarter and the quarter diagonally opposite to it moved inward, while the other two corner moved outward. When the loading was proceeded, the deflection increased dramatically at the loaded quarter. Fig.(7-10) shows the relation between the load and deflection for quarter 1 and 3, while Fig.(7-11) shows quarter 2 and 4 for both practical and numerical results.

7-6 Second case of loading

In the this test the four quarters were loaded to study the symmetrical case, the wiffle tree was hang at sixteen points all over the dome and in equal space, ,see Fig. (7-5-b).

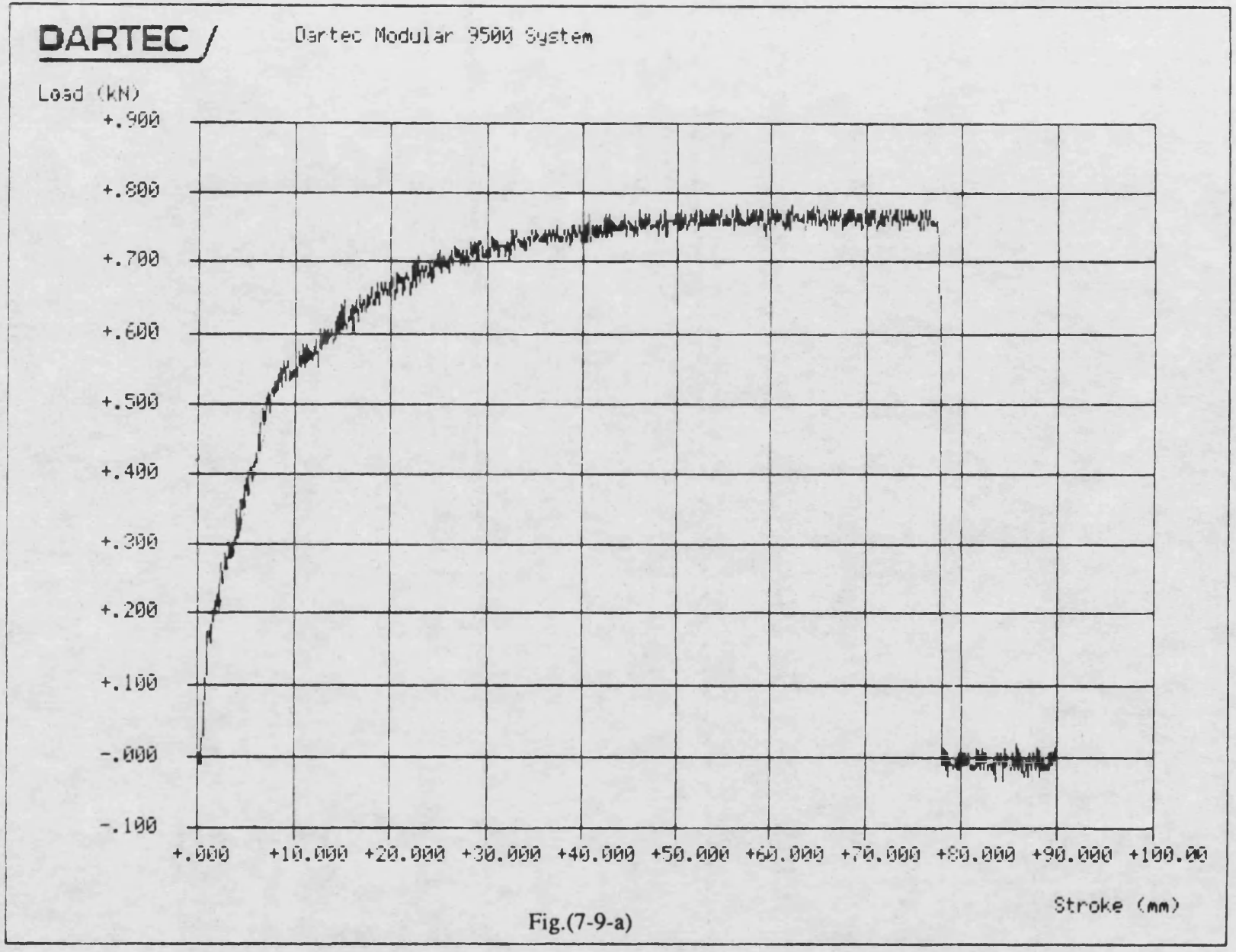

DARTEC
Dartec Modular 9500 System
Load (kN)
+.900
+.800
+.700
+.600
+.500
+.400
+.300
+.200
+.100
-.000
-.100
+.000
+10.000
+20.000
+30.000
+40.000
+50.000
+60.000
+70.000
+80.000
+90.000
+100.00
Stroke (mm)

Fig.(7-9-a)

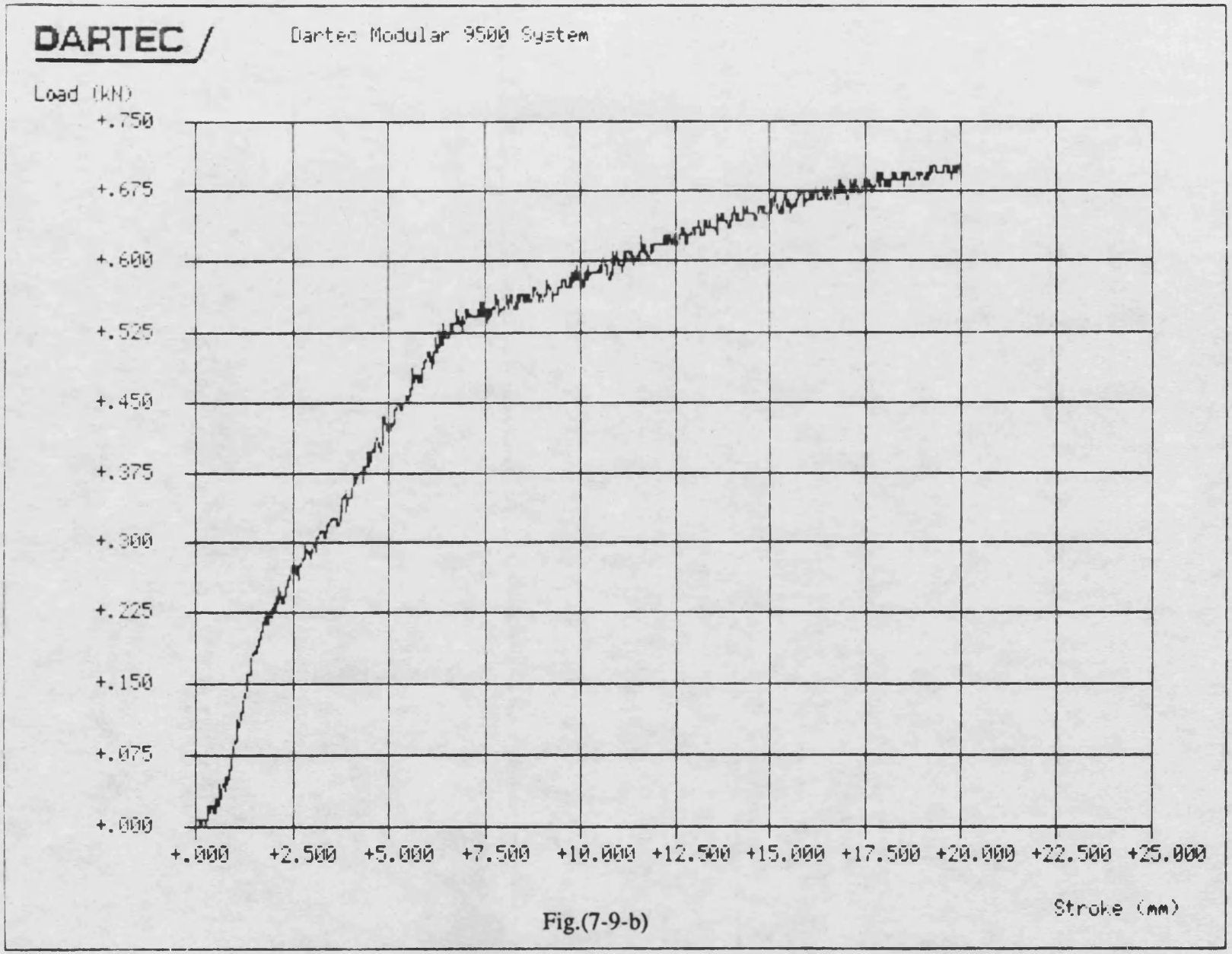
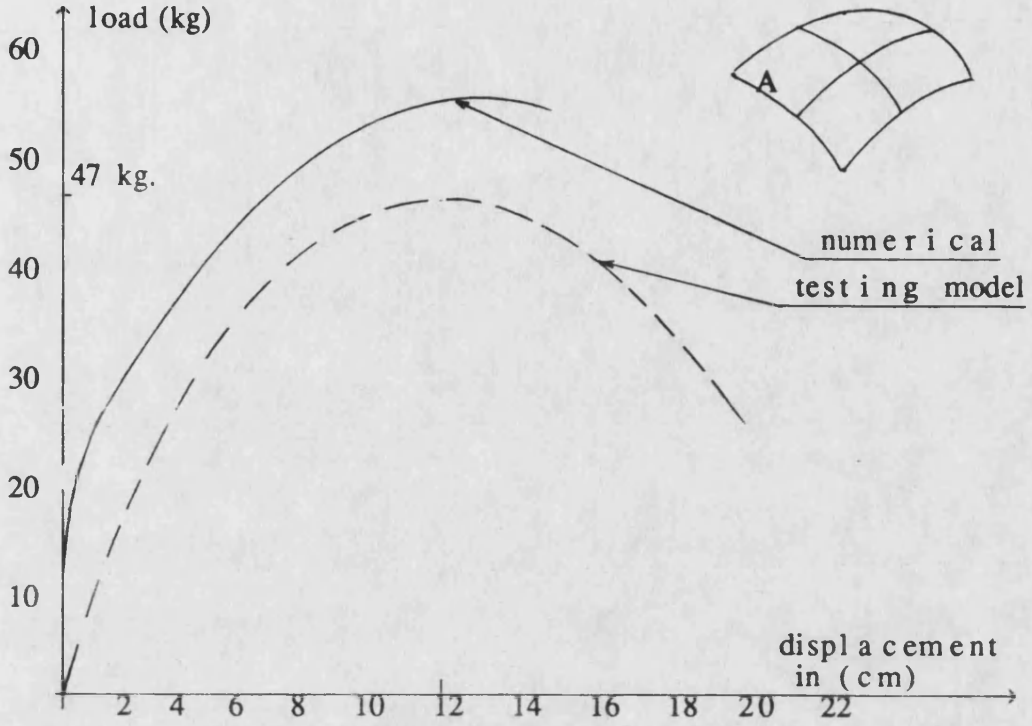
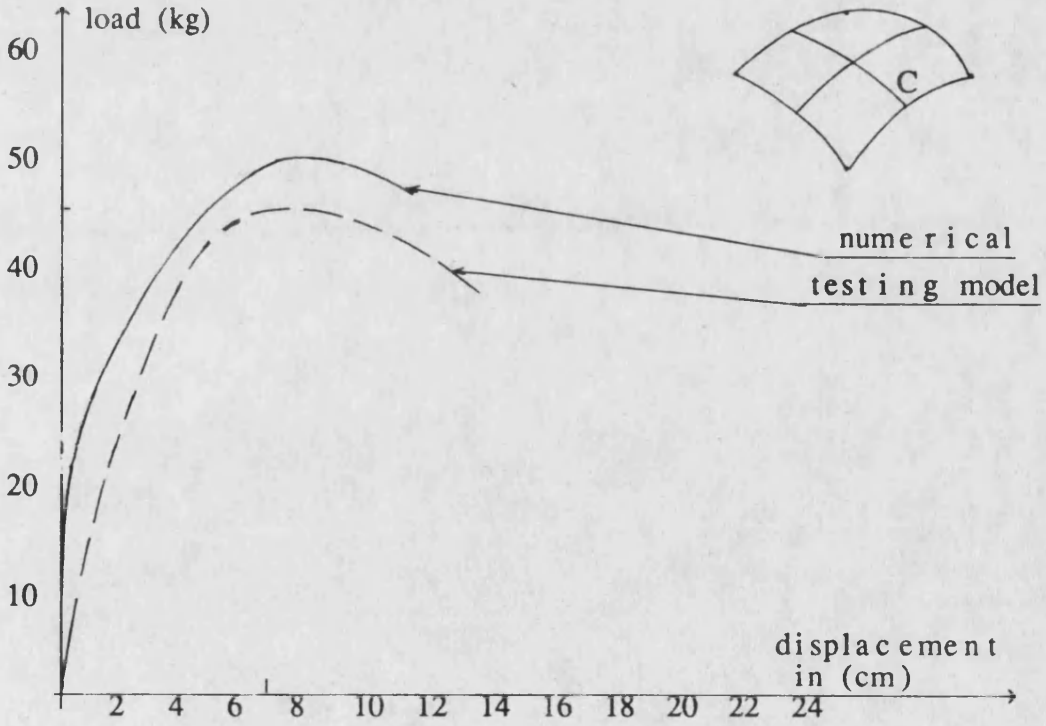

DARTEC
Dartec Modular 9500 System
Load (kN)
+.750
+.675
+.600
+.525
+.450
+.375
+.300
+.225
+.150
+.075
+.000
+.000
+2.500
+5.000
+7.500
+10.000
+12.500
+15.000
+17.500
+20.000
+22.500
+25.000
Stroke (mm)

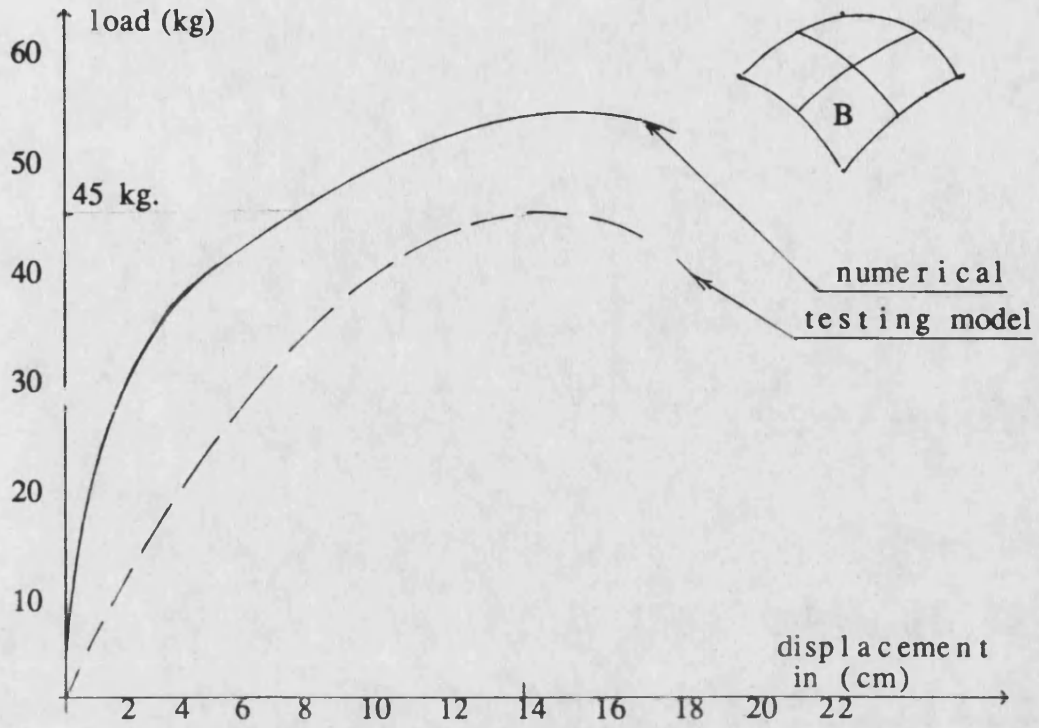
Fig.(7-9-b)

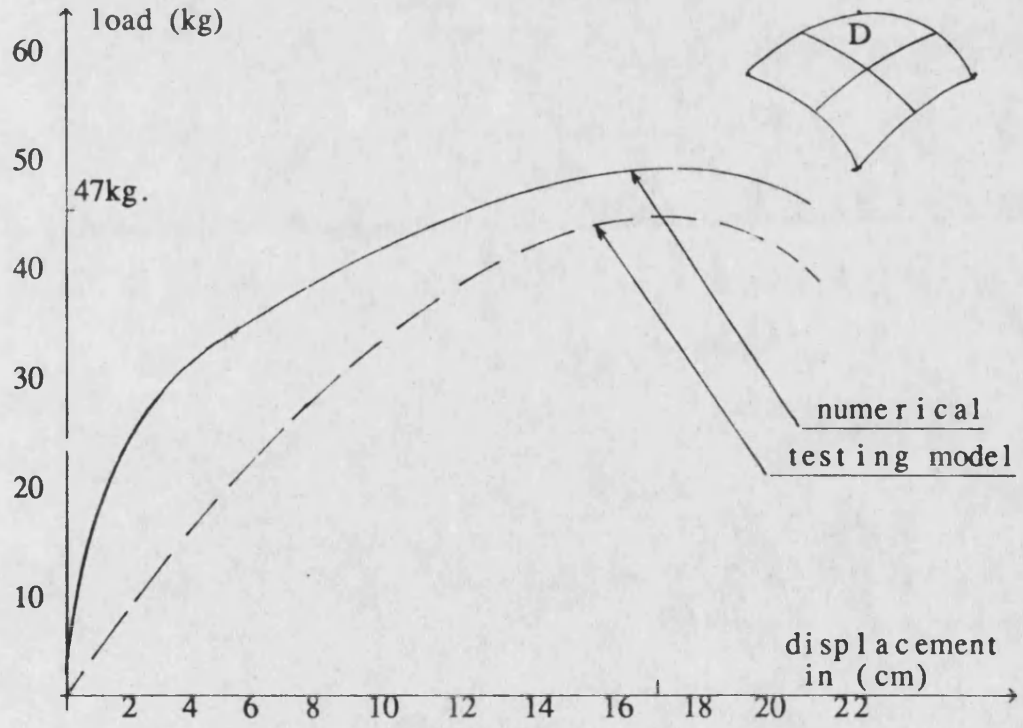




First case (unsymmetrical load)

Fig.(7-10)





First case (unsymmetrical load)

Fig.(7-11)

The grid shell was loaded first using a 10.0 ton jack, and the test was repeated using weights starting from 10.0 kg and increased by 10.0 kg until the surface collapsed. The relation between the applied load and the corresponding deflection was recorded at the middle of each quarter. During the early stage of loading , two diagonal quarters moved inward, while the other two moved outward.

When the load was proceeded, the deflection increased dramatically at the two quarters which moved inward, but prior to total collapse, the whole surface deflected downward. Fig.(7-12) shows the relation between the load and deflection at quarter 1 and 3, while Fig.(7-13) for quarter 2 and 4 for both practical and numerical results.

7-7 Observation and discussion of the test results

During the investigation of the model type, piano wire was the first choice to be used for the surface but the idea was changed for two reasons. First, the piano wires were very stiff to bend, second too much time was required to build all the mesh from a single wire.

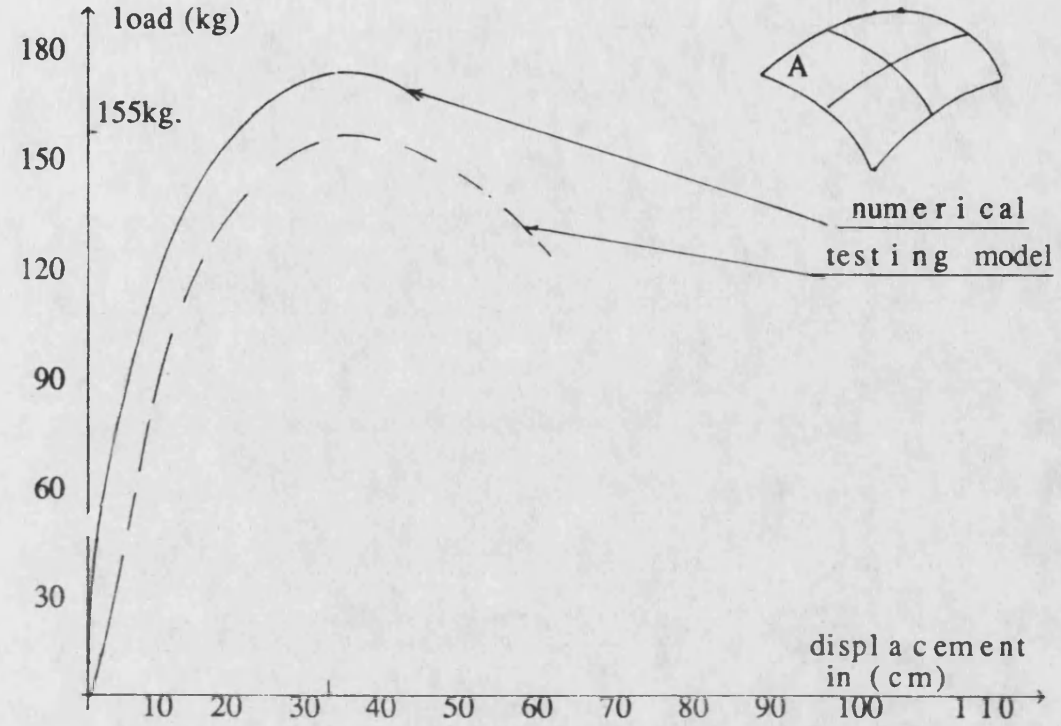
The value of E obtained from the load-displacement curve would appear to be low, this is possibly due to the fact that the sample had a very small diameter (ϕ 1.64 mm), also the wires were not straight due to the mesh form.

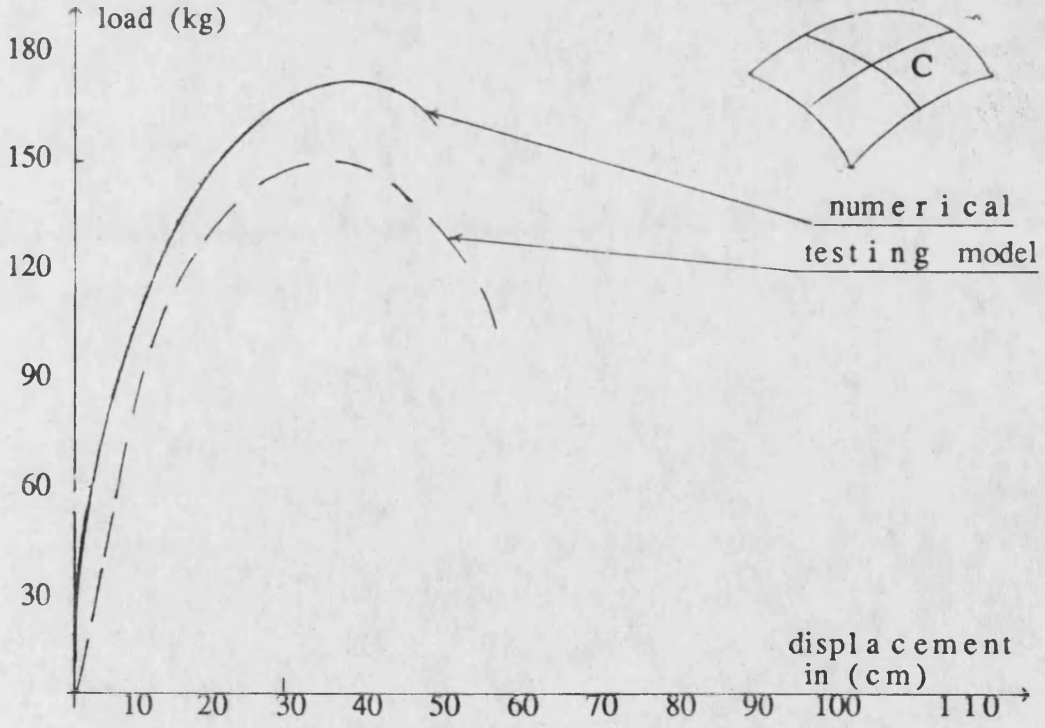
In the first test, the total collapse load was 45.0 kg weight, where the load was manually increased by 5.0 kg every step. In the second case, the loading of the model was repeated again manually because it was found that the total required load was 2% of the total jack capacity and the jack could not provide sufficient sensitive increments of load. The total collapse load was 125.0 kg.

At the beginning of the loading, the two models behaved similarly. But during the loading process, every case behaved differently as shown in Fig.(7-14) and Fig.(7-15).

At 20.0 kg loading in the first test, when the load was removed the shell recovered to its original shape except from only local deformation at the load hanging points. For the second test this occurred at 45.0 kg.

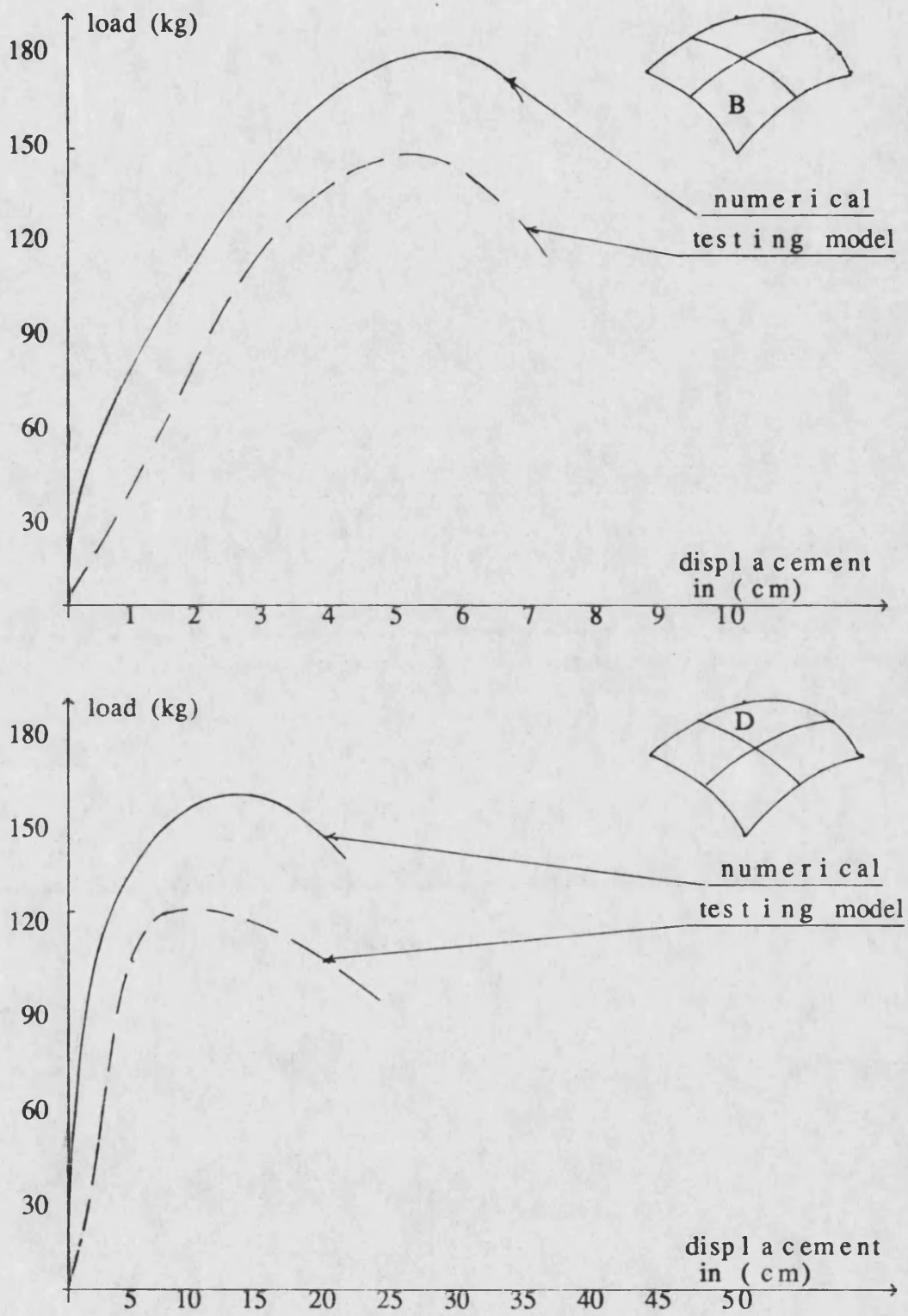
There was no shrinkage or damage at the position of the connection of the wire mesh to the wooden frame and that means no tension forces were produced in that area, also the IPSON 38 is recommended for this type of work.





Second case (symmetric load)

Fig.(7-12)



Second case (symmetric load)

Fig.(7-13)

Fig.(7-14) The unsymmetrical loading

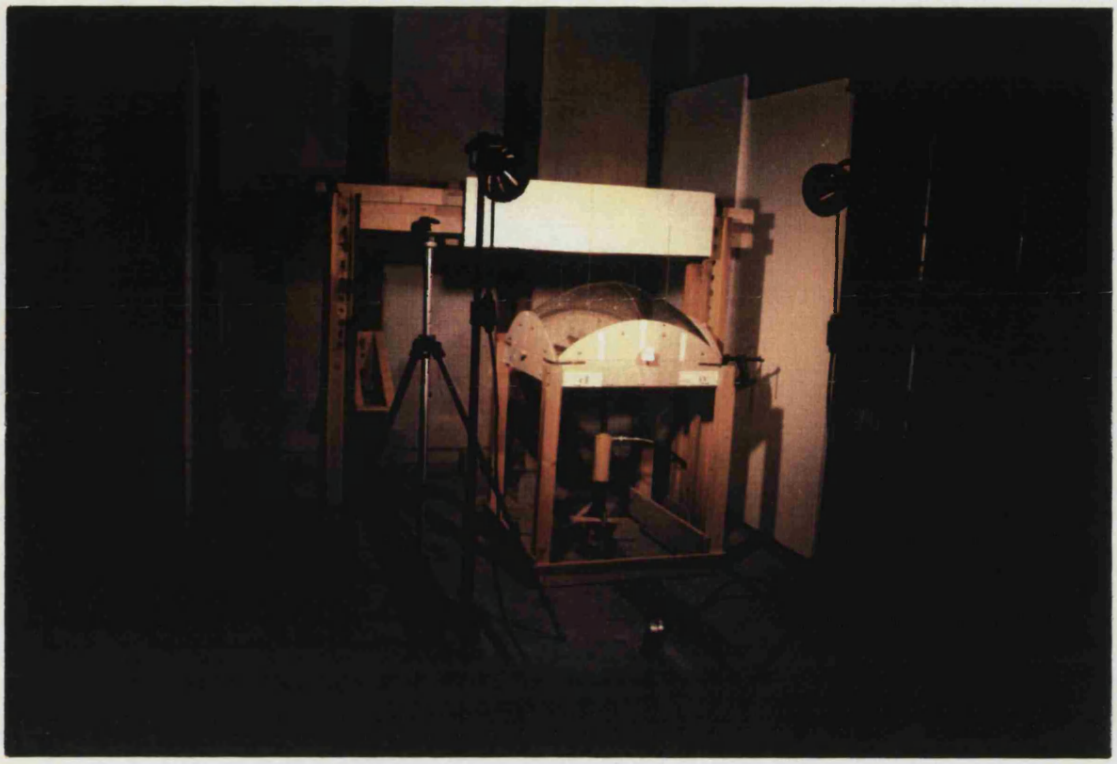


Fig.(7-15) Symmetrical loading

7-8 Conclusion and comparison

From the experimental results and the above discussions, the following conclusions can be drawn;

(1) In the symmetrical case of this dome, the crossing lines ,which divided the dome into four quarters, produced a very high resistance against the deflection up to the full collapse, see Fig.(7-16).

(2) The testing model under unsymmetrical loads behave partially symmetric at the start of the test (two diagonal quarters were moved inward and the other two moved outward).

(3) The deformed shapes obtained from the experimental tests are reasonably consistent with the results of the finite element program (see Figures (6-6),(6-7) and (7-14),(7-15)).

(4) In the first test the collapse load from the computer program was 56.0 kg weight and from the testing model was 45.0 kg.

(5) In the second test the collapse load from the computer program was 150.0 kg weight and from the testing model was 125.0 kg.

(6) The testing model shows a full coincidence with the output results of the computer program.

(7) The agreement between the experimental and the finite element results was successfully demonstrated, therefore the buckling load of grid shells even for those of complex geometry can be predicted.

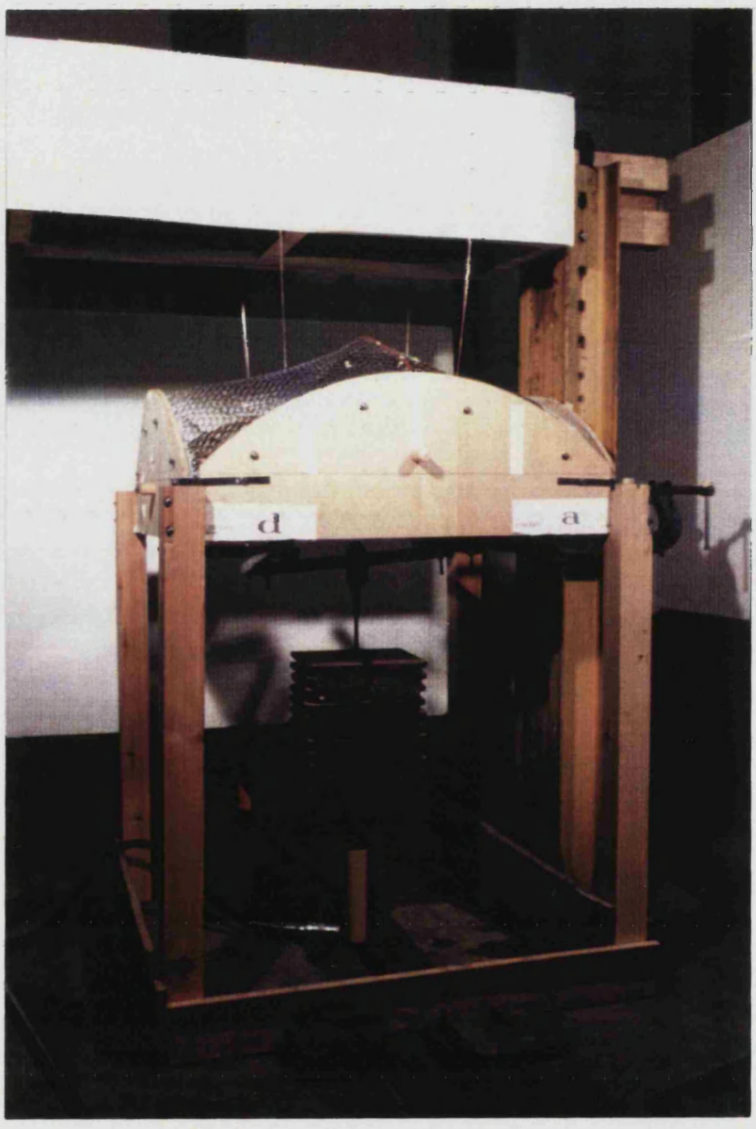


Fig.(7-16) The crossed resistance lines against the deflection

CHAPTER EIGHT

CONCLUSIONS AND FUTURE AREA OF RESEARCH

8-1 Conclusion

The numerical analysis presented in this thesis described the spatial behaviour of the buckling of grid shells. The expression for the normal bending, geodesic bending and torsional moment as well as the membrane forces were derived. The study started with a curve in two dimensions and extended to grid shells in three dimensions.

General conclusions may be drawn from the work as follows;

1- The results obtained from the two-dimensional computer program agreed well with the solutions found in alternative existing analytical procedures and theory of elastica.

2- In the three dimensional analysis, two cases of loading were investigated for the grid shell model, the first case was unsymmetrical where the load was applied in four points at the middle of one quarter, and the second was symmetric where the load was applied in sixteen nodes with equal space all over the shell. The results of the computer program show a good agreement with the behaviour of the physical model tests.

3- A bicubic B-spline element with 48 degrees of freedom was introduced and applied as a new element for the finite element method.

4- Care must be taken during the division of the elements when calculating the axial tension forces. The ratio between the number of constraints and the number of the degrees of freedom in the author's opinion should be approximately 1:2.

8-2 Suggestions for future research

One of the original aims of the work reported in this thesis was to perform general analysis of grid shells. The analysis was successful when the model was divided into a certain number of elements and failed with other, precisely when calculating the axial tension on the surface. Therefore more applications are needed to generate the computer program for any type of data.

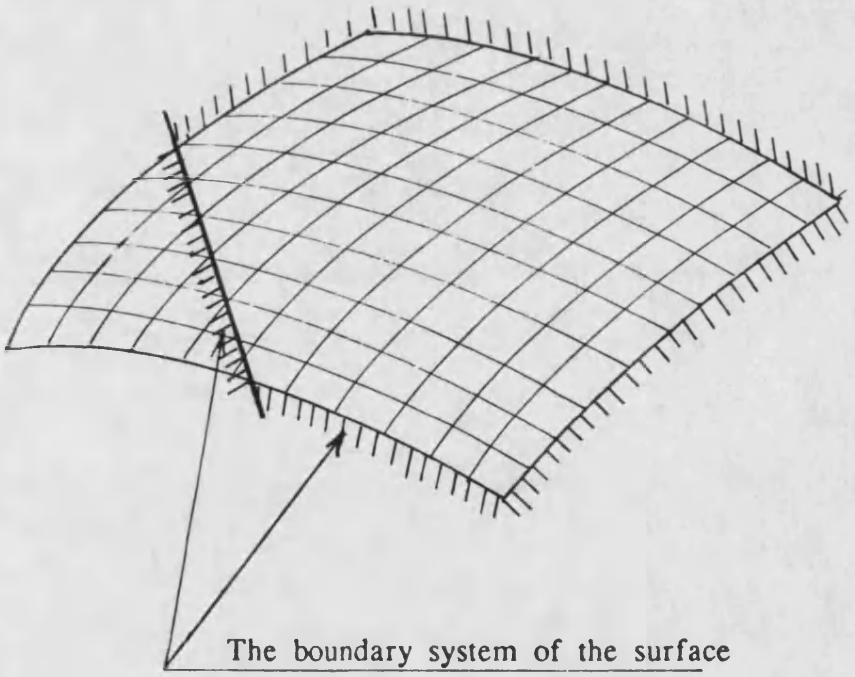
A further numerical development would be needed to study the position of the boundary at the surface, especially, when the boundary is located in skew position to the surface as shown in Fig.(8-1).

The effect of the ties on the lattice shells is very significant, as mentioned in chapter 1. More study is needed to investigate the effect of ties using the computer program.

Creep buckling was studied on columns that have different types of degrees of freedom. However, the effect of creep buckling on grid shells needs further research to calculate the lifetime of such surfaces.

Finally the double layer lattice shell (such as the shell which was used for the Mannheim exhibition centre), needs further investigation to include the effect of shear deformations in the computer program.

A paper, has been submitted to the Institute of Civil Engineering, is inclosed at the end of the appendices.



The boundary system of the surface

Fig.(8-1)

REFERENCES

Allen, H.G. and Bulson, P.S. "Background to buckling" Mc_Graw-hill UK 1980.

Barnes, M. "Review of solution methods of static and Dynamic analysis of tension structures" The institution of structural engineers, The design of air-supported structures, Churchill hall, Bristol, July 1984.

Bickley, W.G. and Gibson, R.E. " Via vector to tensor " The English Universities Press LTD, London, E.C.1, 1965.

Bleich, F. " Buckling strength of metal structures " Mc_Graw Hill book, New York, 1952.

Brebbia, C. and Connor, J. " Geometrically nonlinear finite element analysis " ASCE engineering mechanics division April, PP 463-483, 1969.

Calladine, C.R. " Thin-walled elastic shells analysed by a rayleigh method " Int. J. Solids Structures, Vol. 13 pp. 515-530, 1977.

Calladine, C.R. " Theory of shell structures " Cambridge, The university press, 1983.

Calladine, C.R. "Gaussian curvature and shell structures" The mathematics of surfaces edited by J.A. Gregory Clarendon press, Oxford , proceedings of a conference, 1986.

Chung T.J. " Continuum Mechanics " Prentice Hall, New Jersey, USA, 1988.

Coates, R.C. ,Coutie, M. and Kong, F. " Structural analysis " London, Nelson and sons Ltd , 1972

Connor, J.J., Logcher, R.D. and Shing-ching " Nonlinear analysis of elastic framed structures " ASCE, Journal of the structural division, ST6, PP. 1525-1547, June, 1968.

Coxter, H.S.M. " Introduction to Geometry " John Wiley and Son, N.Y., USA, 1961.

Donnell, L. H. and Wan, C.C., " Effect of Imperfections on buckling of this cylinders and columns under axial compression" Journal of Appl. Mech, ASME, 17(1), 1950.

Faux, I.D. and Pratt, M.J. " Computational Geometry for design and Manufacture, Ellis Horwood, Halsted Press, 1979.

Finnie, L. and Heller, W. R. " Creep of Engineering materials"" by The Mc_Graw Hill Book Company, 1959.

Forsyth, A R " Calculus of variations " Dover, 1960

Ghali, A. " Structural analysis, a unified classical and matrix approach " London, Chapman and Hall ,1978

Gordon, W.J. " Spline-blended surfaces interpolation through curve network's " Journal of Math. and Mech. 18, 10, pp.931-952, 1969.

Gordon, W.J. and Riesenfeld, R.F. " B-spline curves and surfaces " in computer aided geometric design, Academic prss 1974.

Green, A.C. and Zerna, W. " Theoretical elasticity " Oxford 1968

Haisler, W.E. ,Stricklin, J.A. and Stebbins, F.J. " Development and evaluation of solution procedures for geometrically non-linear structural analysis " AIAA journal, Vol.10, No.3, March 1972.

Happold, E and Liddell, W.I. " Timber lattice roof for the Mannheim Bundesgartenschau " The Structural Engineer, No.3, Volume

53, March 1975.

Higgins, T. D. " Effect of creep on column deflections." chapter 20 of Ref. Shanley, F.R., 1951.

Hoff, N.J. " Buckling and stability " Journal of Royal Aeronautical Soc., 58- pp(3-52) 1954.

Hoff, N.J. " Creep buckling " Journal of Aeronautical Quarterly vol. VII pp(1-20) Feb. 1956.

Hoff, N.J. " A survey of the theories of creep buckling " Proceedings of the third U.S. National Congress of applied mechanics pp(29-50), 1958.

Horne, M.R."Elastic-plastic failure loads of plane frames", Proc. Royal Soc., A,274, pp.343-364, 1963.

Hult, J.A.H. "Critical time in creep buckling." Journal of applied mechanics, Trans. ASME, vol. 77 pp(432) 1955.

Hult, J.A.H. " Creep in engineering structures " by Blaisdell Publishing Company, 1966.

Illstion, J.M. " Creep of Concrete " in Creep of Engineering Materials, by C. D. POMEROY, 1978.

IL-10 " Grid shells " Institute of Lightweight Structures (IL) Dir. by Frei Otto, University of Stuttgart, Germany, 1974.

Johanson, A. E., Mathur, V.D. and Henderson, J. " The Creep deflection of magnesium alloy struts " Aircraft Engineering, 28:419 1956.

Karman, L.G.D. and Husue-shen T. " The influence of Curvature on the Buckling Characteristics of Structures " California

Institute of Technology, I.Ae.S., New-York January 1940.

Libove, C. " Creep buckling of columns " J. of the Aeronautical Science, 19- pp(459-467) July 1952.

Lipschutz, M.M. " Differential Geometry " Schaum's outlines of theory and problems McGraw Hill USA, 1969.

Majid, K.I. " Non-linear structures, matrix methods of analysis and design by computers " London, Butterworths, 1970.

Mallet, R.H. and Marcal, P.V. " Finite element analysis of nonlinear structure " Journal of the structural division, Vol. 94, No. ST9, PP. 2081-2105, Sep. 1968.

Marshall, W.T. and Nelson, H.M. " Structures " Pitman Publishing LIT. London, 1969.

Marin, J. " Creep deflection in columns " Journal of applied physics, 10:103, 1947.

Merchant, W." The failure load of rigid jointed frameworks as influenced by stability", The structural engineer 32, 1954.

Odqvist, F. K. G. " Influence of primary creep on column buckling " Journal of applied Mechanics, Trans. ASME vol. 76 pp(295) April 1954.

Pratt, M.J. " Parametric curves and surfaces as used in computer aided design " The mathematics of surfaces edited by J.A. Gregory Clarendon press Oxford, proceedings of a conference, 1984.

Riesenfeld, R.F. " Applications of B-spline approximation to geometric problems of computer aided design. " Ph.D. thesis, Syracuse Univ. 1973.

Roland, Conrad. " The Work of Frei Otto " Longman, This gives a description of Essen lattice shell, 1970.

Ross, A. D. " The Effect of Creep on Instability and Indeterminacy Investigated by Plastic Models." Structural Engineering, 24:413 (1946), 25:179, 1947.

Schoenberg, I.J. " Contributions to the problem of approximation of equidistant data by analytic functions. " Quart. Appl. Math, vol.4, pp.45-99, 1946.

Shanley, F.R. " Weight strength analysis of aircraft structures " Chapters 18 and 19 McGraw Hill Book Company, N.Y. 1952.

Siwgfried, W. " Failure from creep as influenced by state of stress " Journal of Appl. Mechanics, 65:A202 1943.

Stricklin, J.A., Haisler, W.E. and Von Riesemann, W.A. " Geometrically nonlinear analysis by the direct stiffness method " Journal of the structural division, Vol. 97 No. ST9 Sept. 1971, PP. 2299-2314.

Struik, D.J " Lectures on Classical Differential geometry " 2nd Edition, Addison-Wesley, Reading, Massachusetts, 1961.

Struge, D.P. " Improvements to parametric Bicubic surface patches" The mathematics of surfaces edited by J.A.Gregory Clarendon press, Oxford , proceedings of a conference, 1986.

Tezcan, S.S. Ovunc, G. " An iteration method for the non-linear buckling of framed structures " chapter 45 of The international conference on space structures, Univ. of Surrey, Sep. 1966 ed. by R.M.Davies, Blacwell Scientific Pubilcations, 1967.

Tezcan, S.S. and Mahapatra, A.M. " Tangent stiffness matrix for

space frame members " Journal of the Structural Division ASCE, Vol. 95, No. ST6, Hune 1969.

Timoshenko, S. and Gere, J.M. " Theory of elastic stability " McGraw Hill Company 1963.

Timoshenko, S. and Woinowsky-Krieger, S. " Theory of plates and shells " McGraw Hill Company 1970.

Williams, C.J.K. " Form finding and cutting patterns for air supported structures " The Instituting of structural engineers, London, 1980.

Williams, C.J.K. " Defining and designing curved flexible tensile surface structures " The mathematics of surfaces edited by J.A. Gregory, Clarendon press Oxford, proceedings of a conference, 1986.

Williams, C.J.K. " Use of structural analogy in generation of smooth surfaces for engineering purposes " Computer aided design, 1987.

Wright, D.T. " Membrane forces and buckling in reticulated shells " Journal of the Structural Division, ASCE, Vol. 91, No. ST 1, Feb. 1965.

Zienkiewicz, O.C. and Cheung, Y.K. " The finite element method in structural and continuum mechanics " Mc-Graw, London, 1967.

Zienkiewicz, O.C. " Finite element methods in engineering science " New York, McGraw-hill book Co., 1971.

APPENDICES

APPENDIX (A)

Creep buckling

Appendix A-1-a (bar has one degree of freedom)

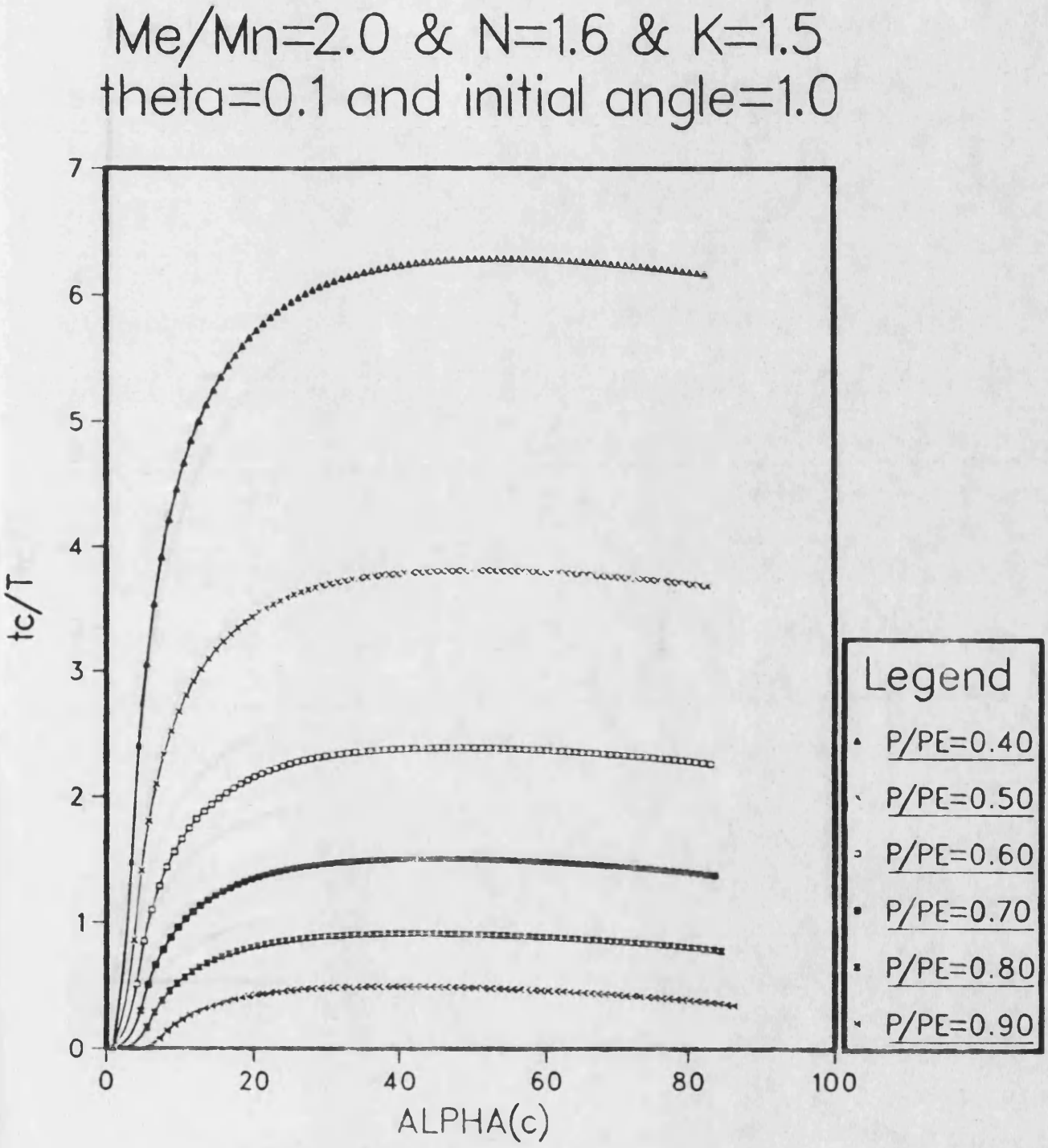


Fig.(A-1)

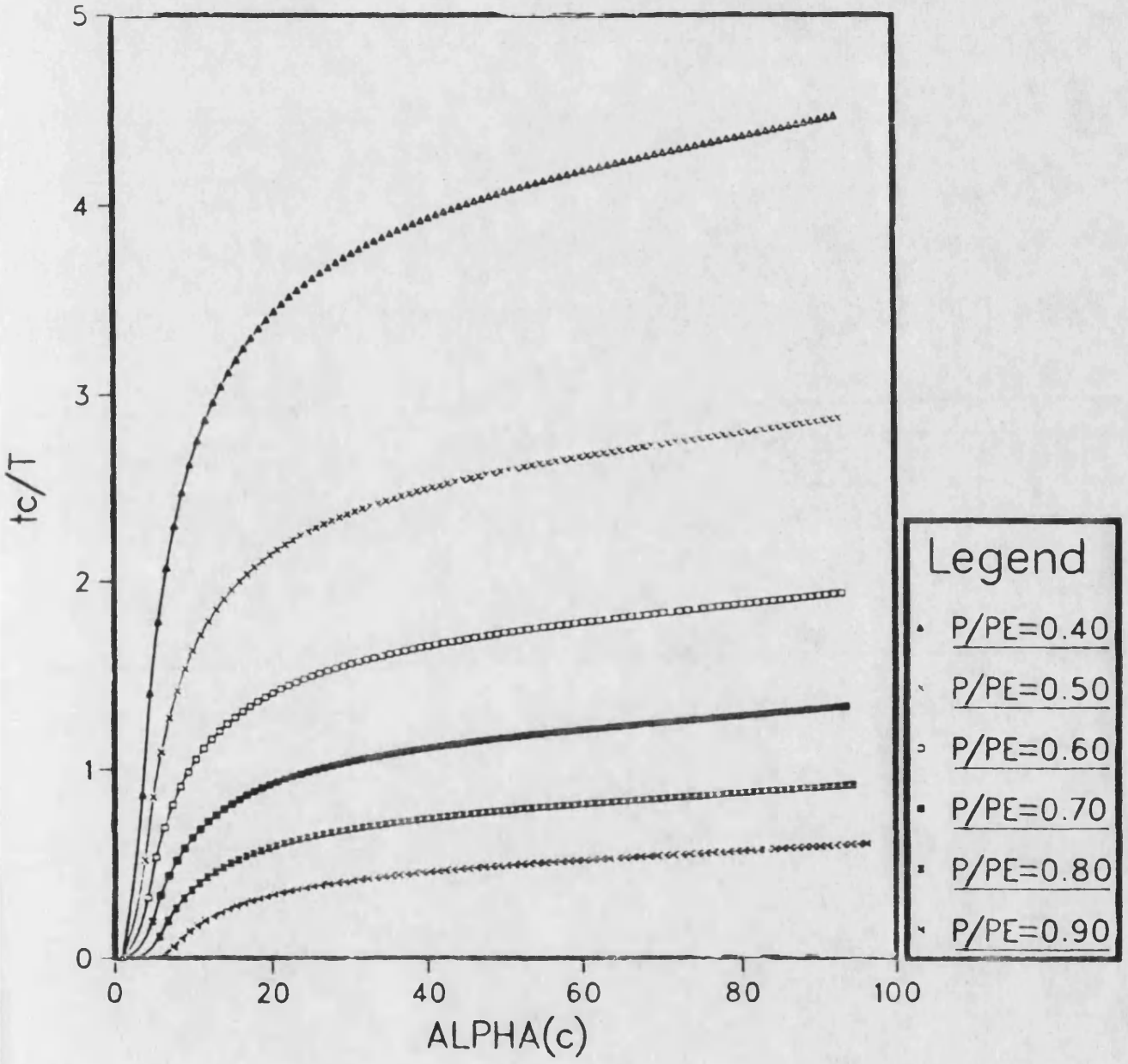

Me/Mn=3.0 & N=1.6 & K=1.5
theta=0.1 and initial angle=1.0
tc/T
ALPHA(c)
0
1
2
3
4
5
0
20
40
60
80
100
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-2)

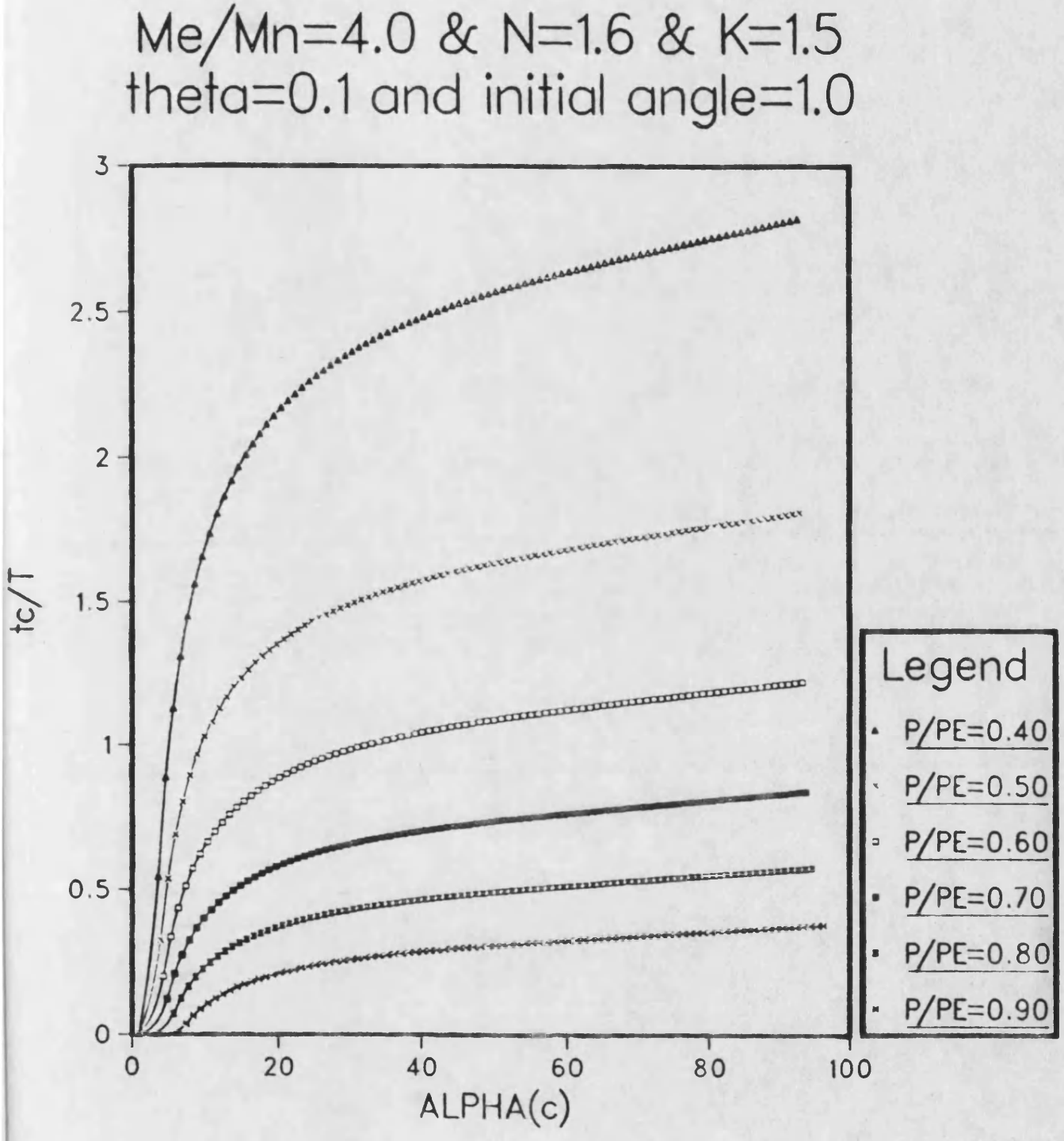

Me/Mn=4.0 & N=1.6 & K=1.5
theta=0.1 and initial angle=1.0
tc/T
ALPHA(c)
0
0.5
1
1.5
2
2.5
3
0
20
40
60
80
100
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-3)

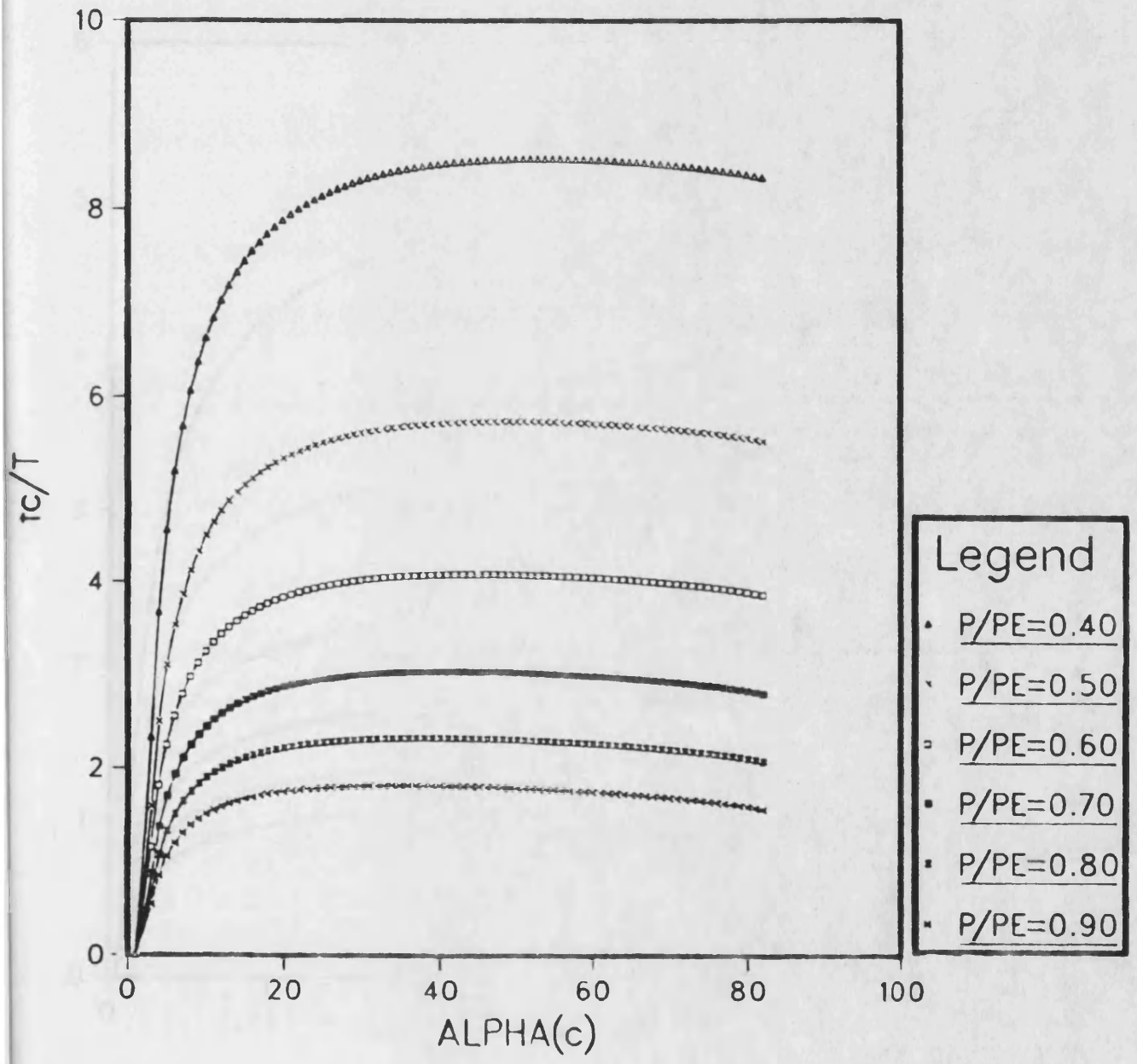

Me/Mn=2.0 & N=1.6 & K=1.5
theta=0.9 and initial angle=1.0
10
8
6
4
2
0
Tc/T
0
20
40
60
80
100
ALPHA(c)
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-4)

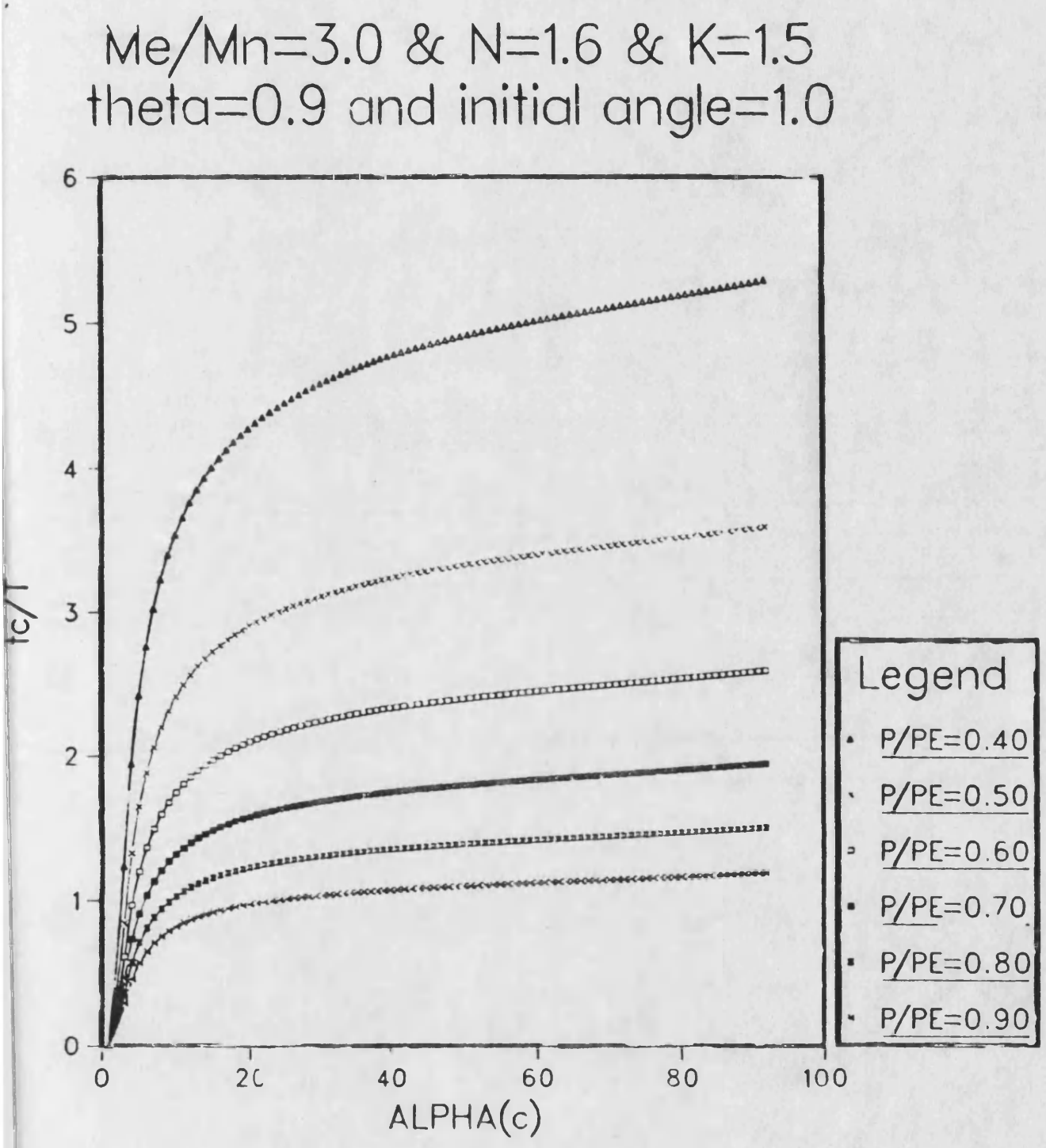

Me/Mn=3.0 & N=1.6 & K=1.5
theta=0.9 and initial angle=1.0
Tc/T
6
5
4
3
2
1
0
0
20
40
60
80
100
ALPHA(c)
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-5)

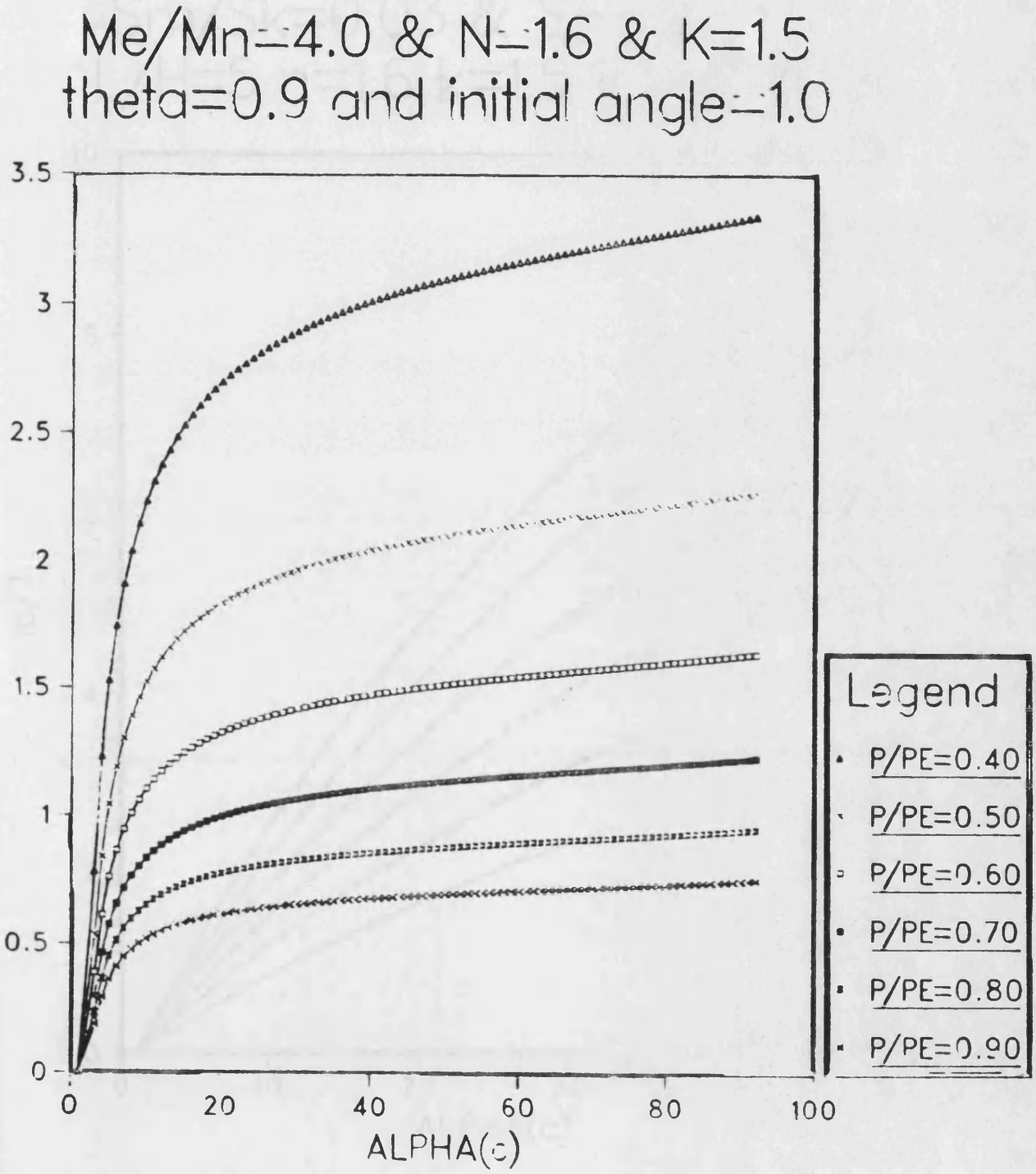

Me/Mn=4.0 & N=1.6 & K=1.5
theta=0.9 and initial angle=1.0
3.5
3
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
ALPHA(c)
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-6)

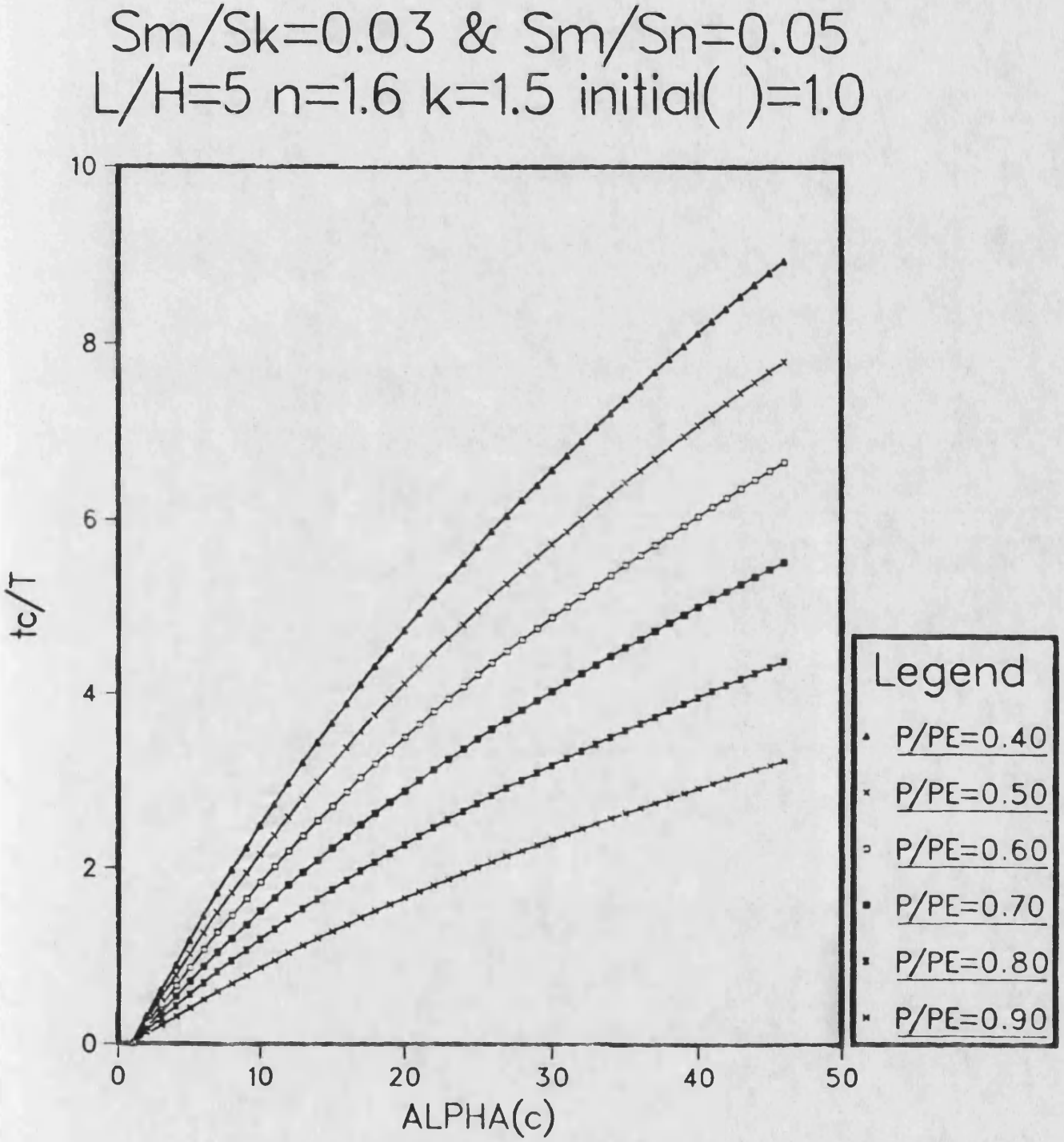

Sm/Sk=0.03 & Sm/Sn=0.05
L/H=5 n=1.6 k=1.5 initial( )=1.0
tc/T
ALPHA(c)
0
2
4
6
8
10
0
10
20
30
40
50
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-7)

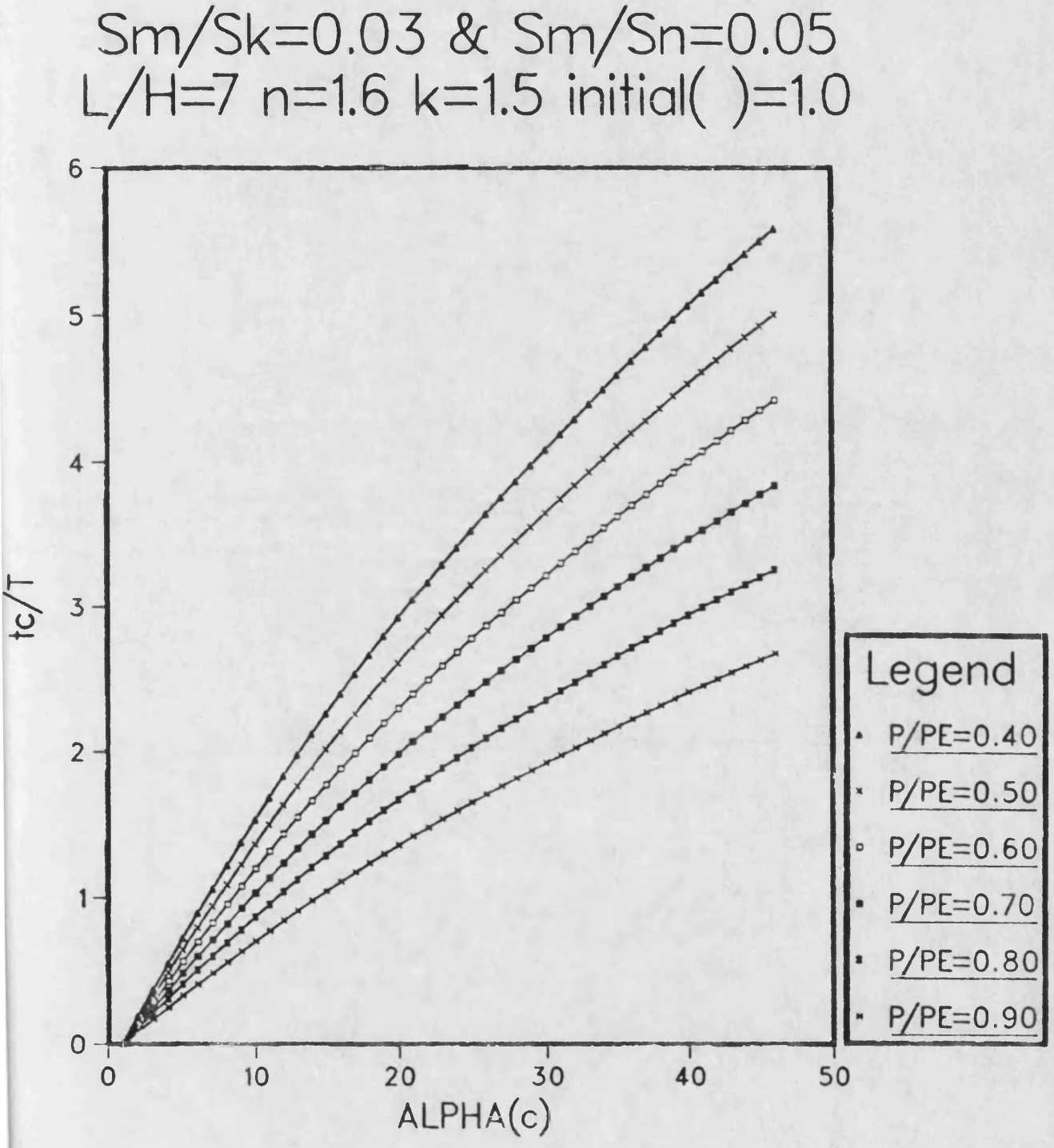

Sm/Sk=0.03 & Sm/Sn=0.05
L/H=7 n=1.6 k=1.5 initial( )=1.0
tc/T
ALPHA(c)
0
1
2
3
4
5
6
0
10
20
30
40
50
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-8)

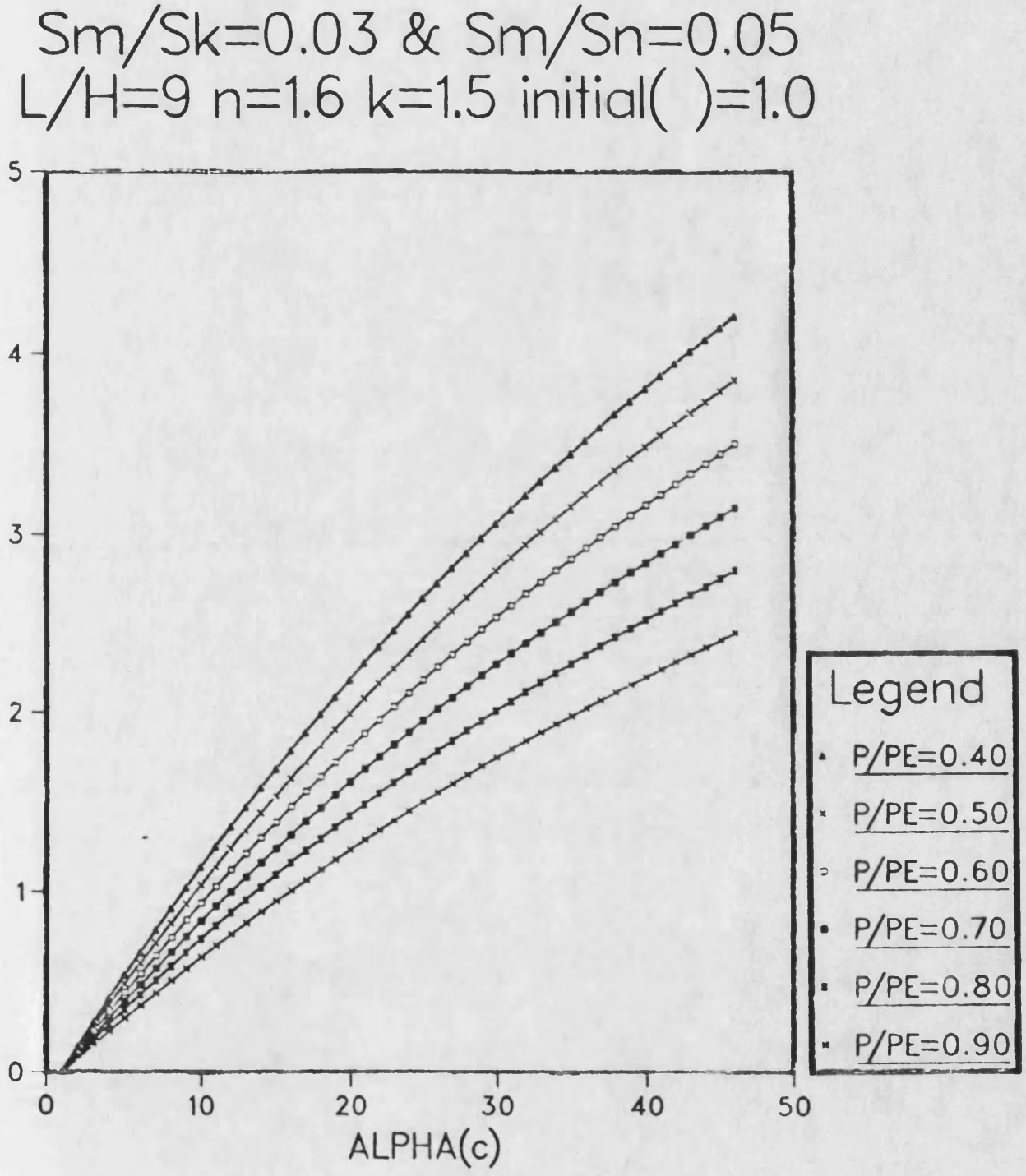

Sm/Sk=0.03 & Sm/Sn=0.05
L/H=9 n=1.6 k=1.5 initial( )=1.0
5
4
3
2
1
0
0
10
20
30
40
50
ALPHA(c)
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-9)

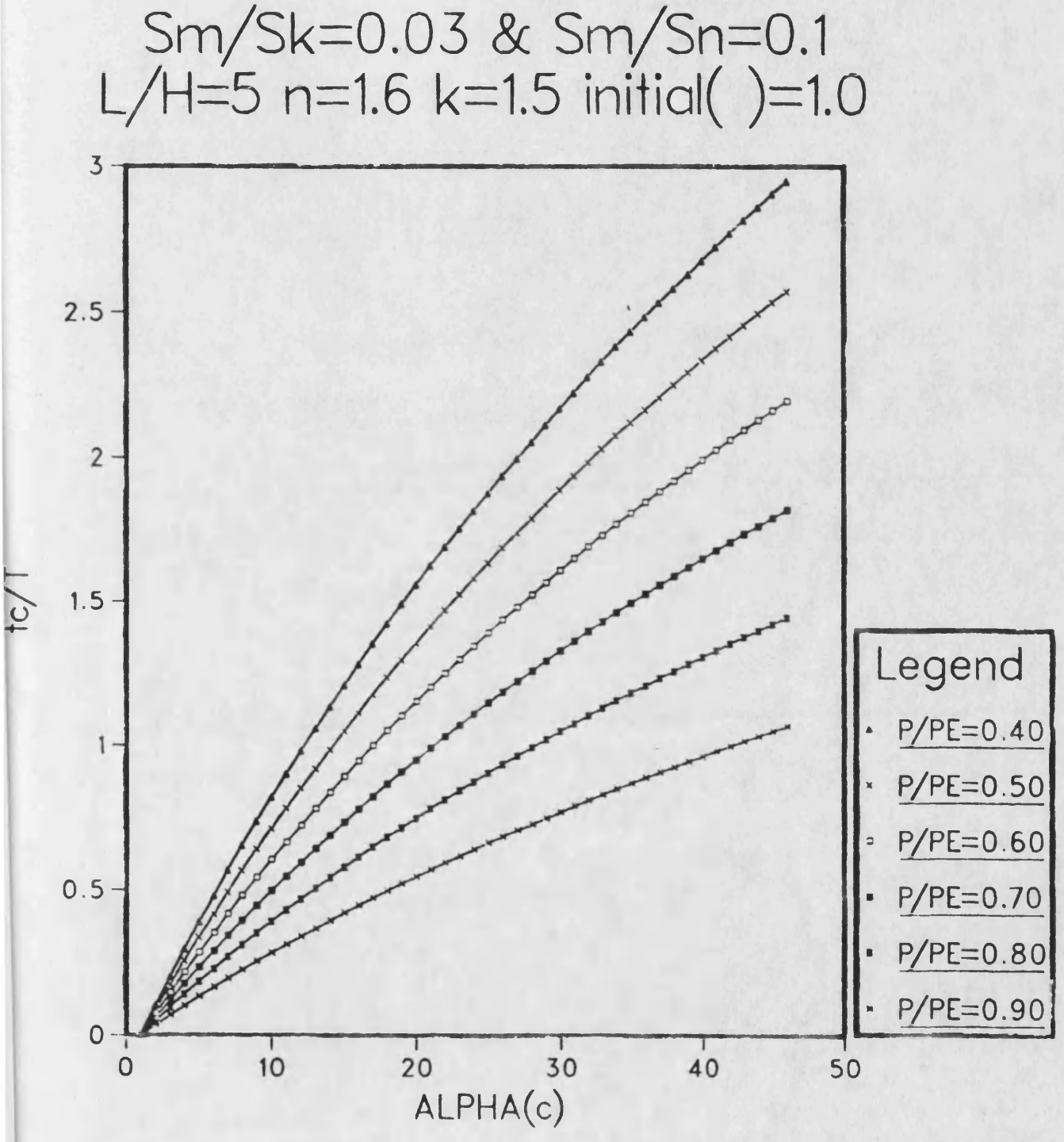

Sm/Sk=0.03 & Sm/Sn=0.1
L/H=5 n=1.6 k=1.5 initial( )=1.0
tc/T
ALPHA(c)
0
0.5
1
1.5
2
2.5
3
0
10
20
30
40
50
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-10)

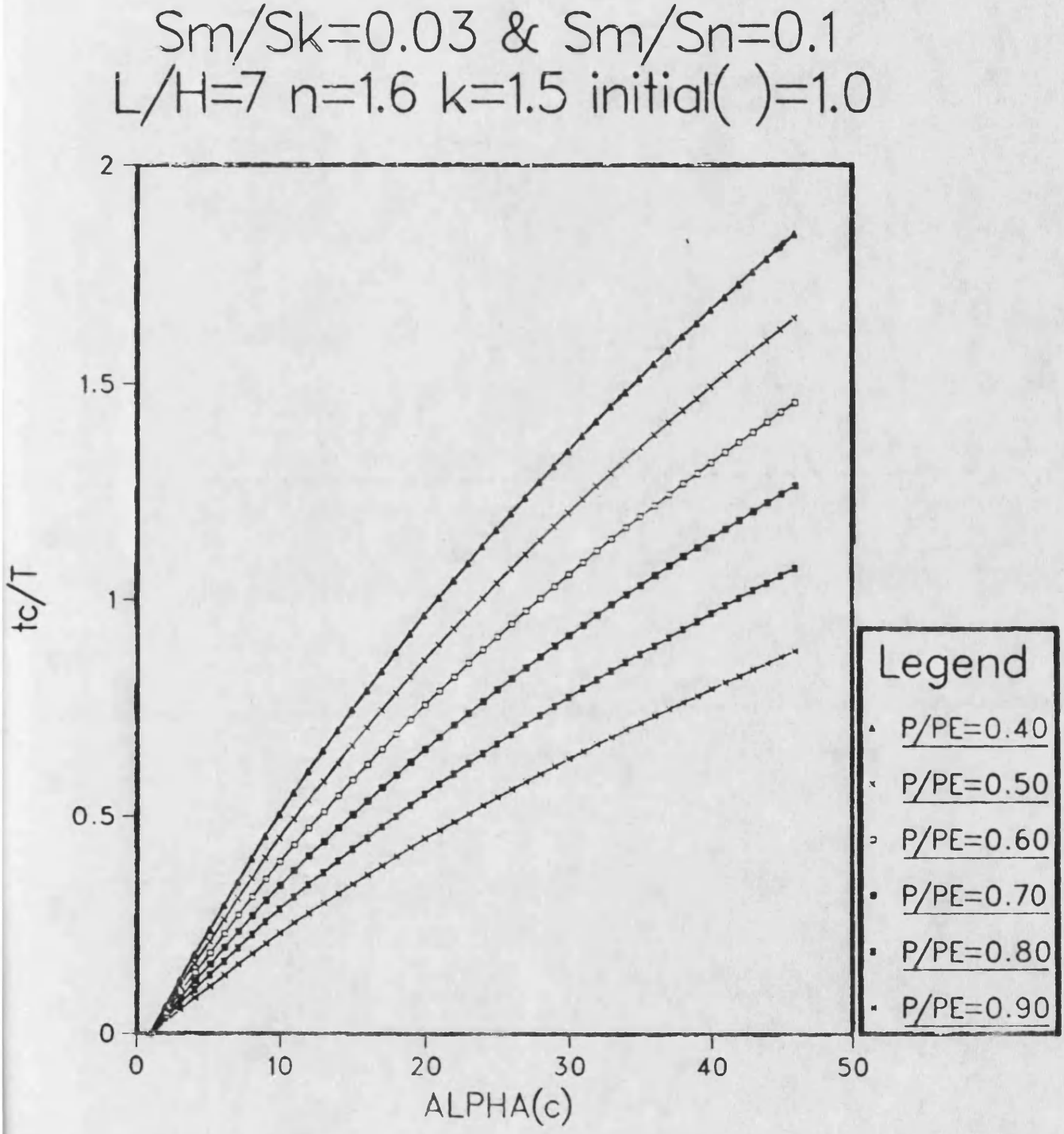

Sm/Sk=0.03 & Sm/Sn=0.1
L/H=7 n=1.6 k=1.5 initial( )=1.0
tc/T
ALPHA(c)
0
0.5
1
1.5
2
0
10
20
30
40
50
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-11)

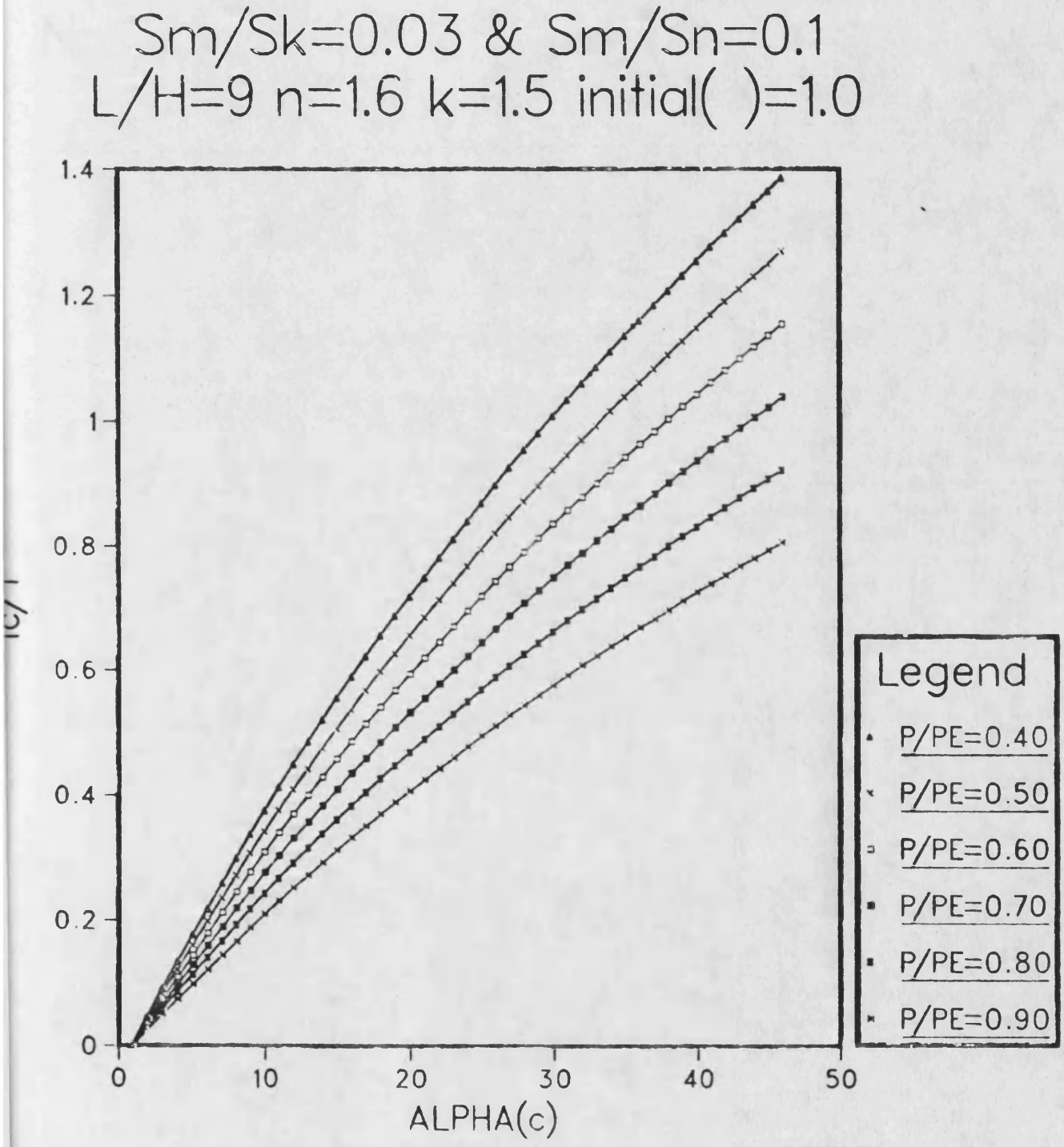

Sm/Sk=0.03 & Sm/Sn=0.1
L/H=9 n=1.6 k=1.5 initial( )=1.0
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
ALPHA(c)
Legend
P/PE=0.40
P/PE=0.50
P/PE=0.60
P/PE=0.70
P/PE=0.80
P/PE=0.90

Fig.(A-12)

Appendix A-1-c (Unstable case)

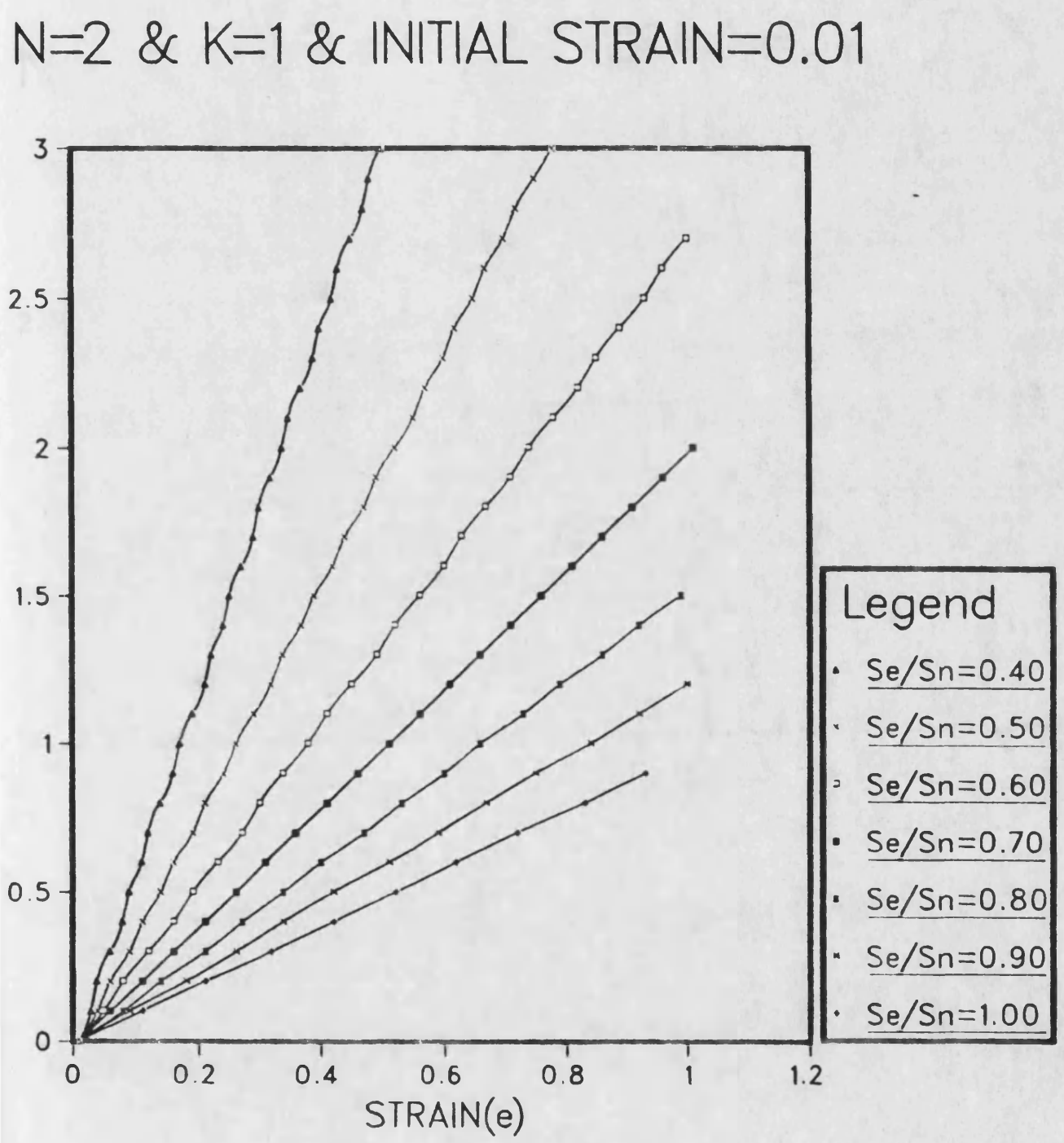


Fig.(A-13)

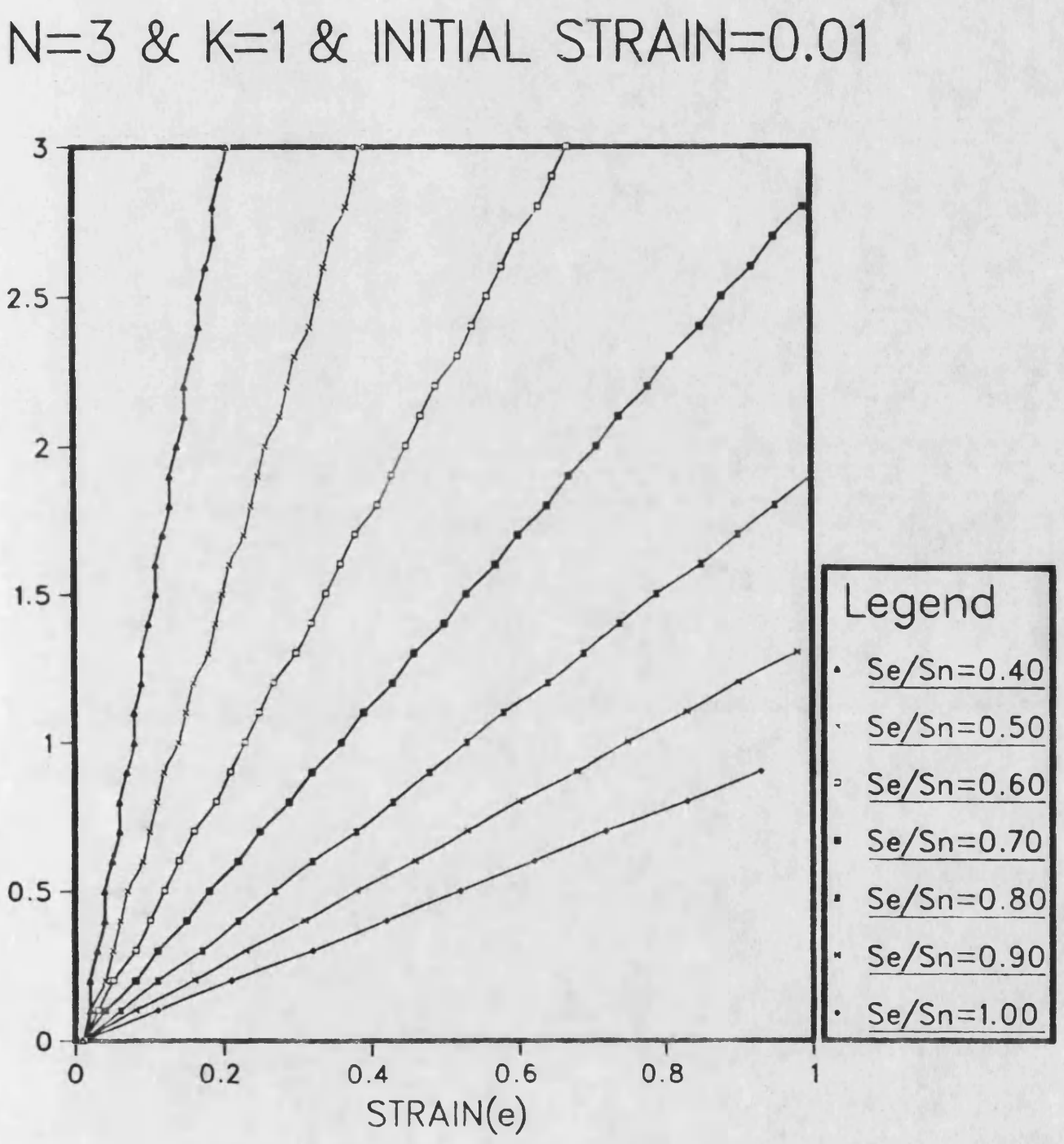

N=3 & K=1 & INITIAL STRAIN=0.01
3
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1
STRAIN(e)
Legend
Se/Sn=0.40
Se/Sn=0.50
Se/Sn=0.60
Se/Sn=0.70
Se/Sn=0.80
Se/Sn=0.90
Se/Sn=1.00

Fig.(A-14)

Appendix A-2 (The computer program lists)

```
C                                CASE No. 1
C
  1   C     ********************************************************
  2   C     PROGRAM TO CALCULATE THE CREEP BUCKLING OF BAR HAS ONE
  3   C     DEGREE OF FREEDOM
  4   C     ********************************************************
  5         INTEGER KOUNT,IFAIL,NOUT,IW(102)
  6         DOUBLE PRECISION W(800),RESULT
  7         DOUBLE PRECISION A,ABSERR,B,EPSABS,EPSREL,EX,EXACT
  8         DOUBLE PRECISION PI, X01AAF,G2,G3,G4,G5,G7,R11
  9   C
 10         EXTERNAL FST
 11         COMMON /TELNUM/ KOUNT,G3
 12   C     ========================================================
 13         INTEGER M,N,MN,I,J,L,K,NK
 14         DOUBLE PRECISION A3,A13,A23,A33,R1,R2,R3,RR,DR,TT
 15   C     ********************************************************
 16         CHARACTER*10 OUTFIL
 17         PRINT *, 'PLEASE, INSERT YOUR OUTPUT FILE---THANK YOU'
 18         READ(5,*) OUTFIL
 19         OPEN(6,FILE=OUTFIL,FORM="FORMATTED")
 20   C     ********************************************************
 21         DO 100 L=1,9,8
 22         G5=L/10.0
 23         DO 100 M=20,40,10
 24         G4=M/10.0
 25         DO 110  N=15,16,1
 26         G3=N/10.0
 27         DO 120 NK=15,16,1
 28         G5=NK/10.0
 29         IF(G3.LE.GK) GOTO 110
 30   C     ========================================================
 31         A3=G3+1
 32         A13=1-G3
 33         A23=(G3+1)/(G3+2)
 34         A33=G3+3
 35   C     ========================================================
 36         WRITE(6,50)G5,G4,G3,GK
 37      50 FORMAT(1X,'THETA=',F6.3,8X,'Me/Mn=',F6.3,8X,'(n)='
 38        *,F6.3),8X,'(K)=',F6.3)
 39   C     ========================================================
 40         DO 200 I=40,90,10
 41         G2=I/100.0
 42         DR=0.0
 43         DO 300 K=10,900,10
 44         GD=K/10.0
 45   C
 46   C     ********************** D01AJF ************************
 47   C
 48         AA=1/(1-(1-G5)*G2)
 49         AD=AA+GD
 50         BD=AD+1.0
 51   C
 52         PI = X01AAF(PI)
 53         EPSABS=0.E0
```

```
      EPSREL=1.E-4
      A =(AD*PI)/180
      B =(BD*PI)/180
      KOUNT = 0
      IFAIL = 1
C
      CALL D01AJF(FST,A,B,EPSABS,EPSREL,RESULT,ABSERR,W,
     *800,IW,102,IFAIL)
C
      SB=SIN(B)
      SA=SIN(A)
      S33=(SB**A13)-(SA**A13)
C
      R1=RESULT
      R2=0.5*G2*(1-G5)*S33/A13
      R3=GK*G5*(G2**GK)*(SB**(GK-G3))-(SA**(GK-G3)))/(GK-G3)
C
      TT=((2*G2*G4)**G3)
      RR=R1-R2-R3
      DD=RR/TT
      DR=DR+DD
C
  300 CONTINUE
  200 CONTINUE
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
C
C
      STOP
      END
C     ==========================================================
C     THE NAG SUBROUTINE (D01AJF)
C
      DOUBLE PRECISION FUNCTION FST(X)
      DOUBLE PRECISION X
      INTEGER KOUNT
      DOUBLE PRECISION G2,G3,G4,R11
C
      COMMON /TELNUM/ KOUNT,G2,G3,G4,R11
C
      KOUNT=KOUNT+1
      FST=2/(SIN(X)**G3)
      RETURN
      END
C     **********************************************************
C
C                              CASE No.2
C
C     **********************************************************
C     PROGRAM TO CALC. THE CREEP BUCKLING OF H-SEC.COL.(2)
C     **********************************************************
      INTEGER KOUNT,IFAIL,NOUT,IW(102)
      DOUBLE PRECISION W(800),RESULT
      DOUBLE PRECISION A,ABSERR,B,EPSABS,EPSREL,EX,EXACT
      DOUBLE PRECISION PI, X01AAF,G2,G3,G4,G5,G7,R11
C
```

```
      EXTERNAL FST,SGN
      COMMON /TELNUM/ KOUNT,G2,G3,G4,R11
C     ========================================================
      INTEGER M,N,MN,I,J,LS,LL
      DOUBLE PRECISION DD,RR,TT
      DOUBLE PRECISION G1,G6,GD,GA,GB,GAA
C     ********************************************************
      CHARACTER*10 OUTFIL
      PRINT *, 'PLEASE, INSERT YOUR OUTPUT FILE---THANK YOU'
      READ(5,*) OUTFIL
      OPEN(6,FILE=OUTFIL,FORM="FORMATTED")
C     ********************************************************
      DO 100 M=3,5,1
      G7=M/100.0
      DO 100 MN=5,10,5
      G6=MN/100.0
      DO 100 N=50,90,20
      G5=N/10.0
C     ========================================================
      WRITE(6,50)G7,G6,G5
   50 FORMAT(8X,'Sm/Sk=',F6.3,X,'Sm/Sn=',F6.3,X,'L/H=',F6.3)
C     ========================================================
      DO 110 I=15,16,1
      G3=I/10.0
      DO 120 J=15,16,1
      G4=J/10.0
      IF(G3.LE.G4) GOTO 110
C
      WRITE(6,60)G3,G4
   60 FORMAT(2X,4H(n)=,F8.2,5X,4H(k)=,F8.2)
C     ========================================================
      DO 200 JJ=10,50,20
      GA=JJ/10.0
      WRITE(6,51)GA
   51 FORMAT(6X,'THE INITIAL ANGLE =',F6.2)
C
C    ******************** P/PE & tc/T **********************
C
      WRITE(6,55)
   55 FORMAT(10X,4HP/PE,11X,4Htc/T,10X,7HALFA(c))
C
      DO 250 LS=40,90,10
      G2=LS/100.0
      DO 300 LL=10,200,10
      G1=LL/100.0
      DO 400 KK=20,500,20
      GD=KK/10.0
C
C    ********************** D01AJF ***********************
C
      PI = X01AAF(PI)
      AD=GA
      BD=AD+GD
      A =(AD*PI)/180
      B =(BD*PI)/180
      TT=2*G1*((G5/PI)**2)*(G6**(G3))
      R11=2*G4*((G5/PI)**2)*(G7**(G4))
```

```
66    C
67          EPSABS=0.E0
68          EPSREL=1.E-4
69          KOUNT = 0
70          IFAIL = 1
71    C
72          CALL D01AJF(FST,A,B,EPSABS,EPSREL,RESULT,ABSERR,W,
73         *800,IW,102,IFAIL)
74    C
75          RR=RESULT
76          DD=TT-RR
77          IF(DD.GT.0.01) GOTO 400
78          GO TO 300
79    C
80        400 CONTINUE
81        300 CONTINUE
82        250 CONTINUE
83        200 CONTINUE
84        120 CONTINUE
85        110 CONTINUE
86        100 CONTINUE
87    C
88    C
89          WRITE(6,80) A,B,EPSABS,EPSREL
90          WRITE(6,81) RESULT,ABSERR,KOUNT,IW(1),IFAIL
91         80 FORMAT (1H,2X,1HA,6X,31H- LOWER LIMIT OF INTEGRATION=,
92           *E10.3/1H,2X,1HB,6X,31H- UPPER LIMIT OF INTEGRATION = ,
93           *E10.3/1H,2X,39HEPSABS - ABSOLUTE ACCURACY REQUESTED =,
94           *E9.2/1H,2X,39HEPSREL - RELATIVE ACCURACY REQUESTED = ,
95           *E9.2/)
96         81 FORMAT (1H,2X,41HRESULT-APPROXIMATION TO THE INTEGRAL=,
97           *E14.5/1H,2X,42HABSERR-ESTIMATE OF THE ABSOLUTE ERROR=,
98           *E10.3/1H,2X,42HKOUNT-NO. OF THE FUNCTION EVALUATIONS=,
99           *I4/1H,2X,43HIW(1) - ELEMENTS OF REAL WORKSPACE USED=,
100          *I4/1H,2X,21HIFAIL - ERROR FLAG = ,I4/)
101          P1=(2/G6)**G3
102          G1=RESULT*P1
103   C
104         STOP
105         END
106   C     ========================================================
107   C     NAG SUBROUTINE (D01AJF)
108   C
109         DOUBLE PRECISION FUNCTION FST(X)
110         DOUBLE PRECISION X
111         DOUBLE PRECISION G2,G3,G4,R11
112         INTEGER KOUNT
113   C
114         COMMON /TELNUM/ KOUNT,G2,G3,G4,R11
115   C
116         KOUNT=KOUNT+1
117         FST=(1-G2-R11*((1+X)**(G4-1)))/(((1+X)**(G3))
118        *-(SGN(1-X)*((1-X)**G3)))
119   C
120         RETURN
121         END
122   C
```

```
      DOUBLE PRECISION FUNCTION SGN(X)
      DOUBLE PRECISION X
      IF(X.LT.0.00) THEN
      SGN=-1.00
      ELSE IF(X.EQ.0.00) THEN
      SGN=0.00
      ELSE
      SGN=1.00
      ENDIF
      RETURN
      END
C     ********************************************************

C
C                              CASE No.3
C
C     ********************************************************
C     PROGRAM TO CALCU.THE CREEP BUCKLING OF UNSTABLE COLUMN
C     ********************************************************
      INTEGER KOUNT,IFAIL, NOUT,M,N,L,K,I,J,JJ,NK
C
      DOUBLE PRECISION W(800)
      DOUBLE PRECISION X01AAF,RESULT
      DOUBLE PRECISION A,ABSERR,B,EPSABS,EPSREL,EX,EXACT
C
      EXTERNAL FST
      COMMON /SELNUM/ KOUNT,R11,R22,G3,GK
C     ========================================================
      DOUBLE PRECISION PI,G2,G3,G4,G5,G6,GK,AA,GD
      DOUBLE PRECISION R11,R22,RR,DD,DR,TT
C     ********************************************************
      CHARACTER*10 OUTFIL
      PRINT *, 'PLEASE, INSERT YOUR OUTPUT FILE___THANK YOU'
      READ(5,*) OUTFIL
      OPEN(6,FILE=OUTFIL,FORM='FORMATTED')
C     ********************************************************
      DO 100 L=1,9,8
      G5=L/10.0
      DO 100 M=20,40,10
      G6=M/10.00
      DO 110 N=15,16,1
      G3=N/10.0
      DO 120 NK=15,16,1
      G4=NK/10.0
      IF(G3.LE.G4) GOTO 110
      GK=G4-1
C     ========================================================
      WRITE(6,50)G5,G6,G3,G4
50    FORMAT(1X,'THETA=',F6.2,8X,'Me/Mn=',F6.2,8X,' (n)='
     *,F6.3,8X,' (k)=',F6.3)
C
C     ************************* P & T **********************
C
      WRITE (6,76)
76    FORMAT(10X,5HP/PE*,10X,4HTC/T,10X,7HALFA(C))
      WRITE(6,566)
  566 FORMAT(1H,'input data.')
```

```
C
      DO 200 I=40,90,10
      G2=I/100.0
C
      WRITE(6,233)G2
  233 FORMAT(1H,'"P/PE=',F4.2,'"')
      WRITE(6,234)
  234 FORMAT(8X,'1.000',8X,'0.000')
      DR=0.0
C
      DO 300 K=10,800,10
      GD=K/10.0
C
C     *************************** D01AJF ********************
C
      R11=(1-G5)*G2
      R22=G4*G5*(G2**G4)
C
      AA=1/(1-(1-G5)*G2)
      AD=AA+GD
      BD=AD+1.0
C
      PI=X01AAF(PI)
      EPXABS=0.E0
      EPSREL=1.E-04
      A=(AD*PI)/180
      B=(BD*PI)/180
      KOUNT=0
      IFAIL=1
C
      CALL D01AJF(FST,A,B,EPSABS,EPSREL,RESULT,ABSERR,W,
     *800,IW,102,IFAIL)
C
      RR=RESULT
C
      TT=((2*G2*G6)**G3)
      DD=RR/TT
      DR=DR+DD
C
300   CONTINUE
200   CONTINUE
C
      WRITE(6,225)
  225 FORMAT(1H,'end of data.')
C
120   CONTINUE
110   CONTINUE
100   CONTINUE
C
      STOP
      END
C
C     ********************* SUBROUTINE **********************
C
      DOUBLE PRECISION FUNCTION FST(X)
      DOUBLE PRECISION X
C
```

```
 99          INTEGER KOUNT
100          DOUBLE PRECISION COS,TAN
101          DOUBLE PRECISION R11,R22,G3,GK
102    C
103          COMMON /SELNUM/ KOUNT,R11,R22,G3,GK
104          KOUNT=KOUNT+1
105    C
106          FST=(2*COS(X)-(R11*(1/COS(X)**2))-(R22*(TAN(X)**GK)*
107         *(1/(COS(X)**2))))/(TAN(X)**G3)
108    C
109          RETURN
110          END
111    C     ********************************************************
```

APPENDIX (B)

Two-dimensions analysis

Appendix B-1-a

$DI(1,1)=DI(2,2)=u-u^2+\frac{u^3}{3}$

$DI(3,3)=DI(4,4)=4u-6u^2+3u^3$

$DI(5,5)=DI(6,6)=u-3u^2+3u^3$

$DI(7,7)=DI(8,8)=\frac{u^3}{3}$

$DI(1,3)=DI(3,1)=DI(2,4)=DI(4,2)=-2u+\frac{5u^2}{3}-u^3$

$DI(1,5)=DI(5,1)=DI(6,2)=DI(2,6)=u-2u^2+u^3$

$DI(1,7)=DI(7,1)=DI(2,8)=DI(8,2)=\frac{u^2}{2}-\frac{u^3}{3}$

$DI(3,5)=DI(5,3)=DI(4,6)=DI(6,4)=-2u+\frac{u^2}{2}-3u^3$

$DI(3,7)=DI(7,3)=DI(4,8)=DI(8,4)=-u^2+u^3$

$DI(5,7)=DI(7,5)=DI(6,8)=DI(8,6)=\frac{u^2}{2}-u^3$

Appendix B-1-b

$B(1,1)=1-4u+6u^2-4u^3+u^4$

$B(1,2)=B(2,1)=4u-11u^2+10u^3-3u^4$

$B(1,3)=B(3,1)=-1+6u-8u^3+3u^4$

$B(1,4)=B(4,1)=-u^2+2u^3-u^4$

$B(2,2)=16u^2-24u^3+9u^4$

$B(2,3)=B(3.2)=-4u-5u^2+18u^3-9u^4$

$B(2,4)=B(4,2)=-4u^3+3u^4$

$B(3,3)=1+4u-2u^2-12u^3+9u^4$

$B(3,4)=B(4,3)=u^2+2u^3-3u^4$

$B(4,4)=u^4$

Appendix B-1-c

$$FAE1 = \frac{6Q_2}{6x1} =$$

$$\frac{\alpha}{8}\left[\frac{1}{L^4}\left(\{P\}^T[BI_1]\{P\}.2\{P\}_1^T[B]\{P\}+\{P\}^T[B]\{P\}.2\{P\}_1^T[BI_1]\{P\}\right)\right.$$

$$-\left(\{P\}^T[BI_2]\{P\}.2\{P\}_1^T[BD_1]\{P\}+\{P\}^T[BD_2]\{P\}.2\{P\}_1^T[BI_2]\{P\}\right)$$

$$+\left(\{P\}^T[BI_3]\{P\}.2\{P\}_1^T[BD_2]\{P\}+\{P\}^T[BD_2]\{P\}.2\{P\}_1^T[BI_3]\{P\}\right)$$

$$-\left(\{P\}^T[BI_4]\{P\}.2\{P\}_1^T[BD_3]\{P\}+\{P\}^T[BD_3]\{P\}.2\{P\}_1^T[BI_4]\{P\}\right)$$

$$+\left(\{P\}^T[BI_5]\{P\}.2\{P\}_1^T[BD_4]\{P\}+\{P\}^T[BD_4]\{P\}.2\{P\}_1^T[BI_5]\{P\}\right)$$

$$\left.+\frac{2}{L^2}\left(2\{P\}_1^T[BI_1]\{P\}\right)\right]$$

Appendix B-1-d

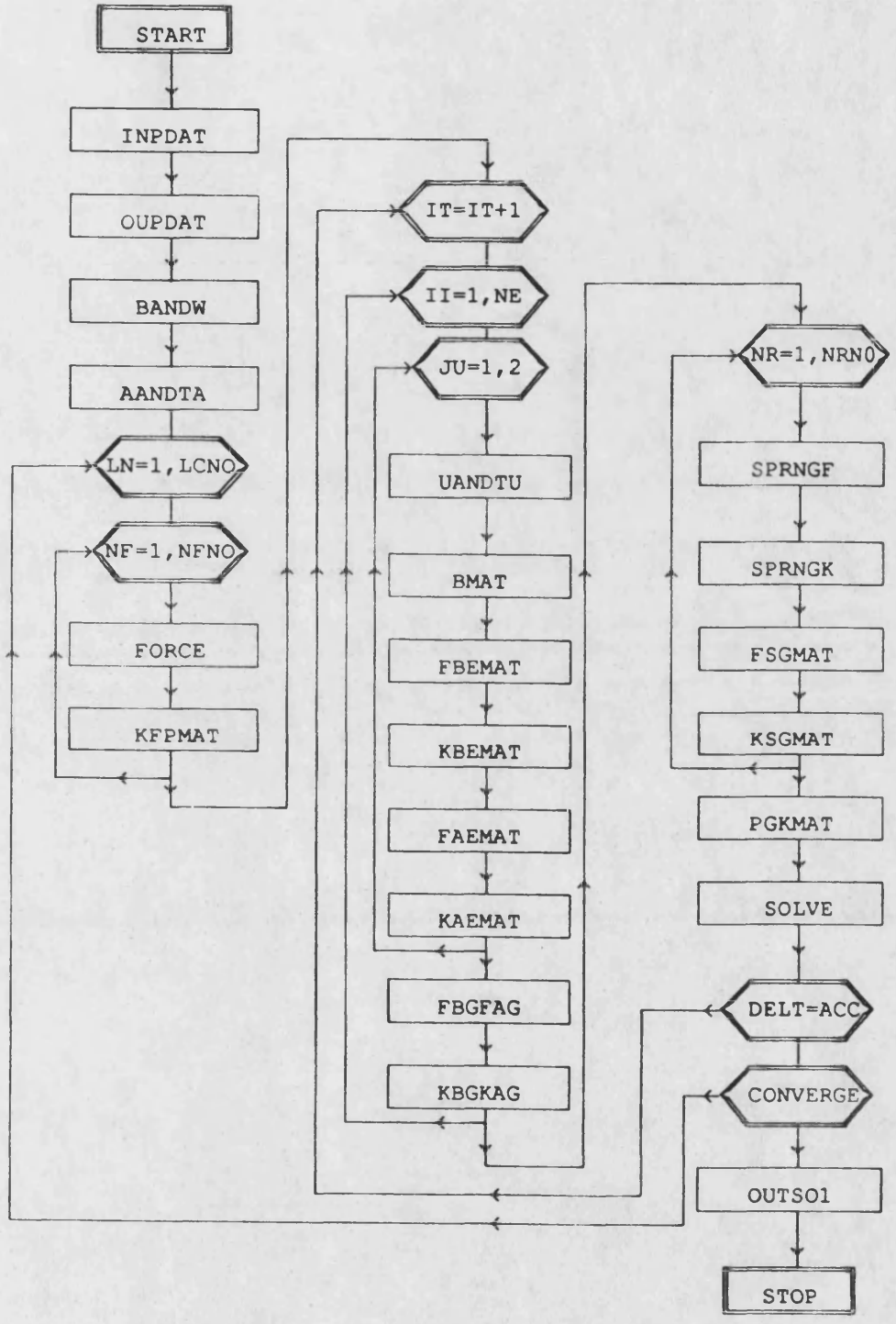
$$KAE1=\frac{6Q_2^2}{6x1\ 6x1}=\frac{\alpha}{8}\left\{\frac{1}{L^4}\left[\left(2\{P\}^T[BI_1]\{P\}.B(1,1)+4\{P\}_1^T[B]\{P\}.\{P\}_2^T[BI_1]\right.\right.\right.$$
$$\left.+4\{P\}_1^T[BI_1]\{P\}.\{P\}_2^T[B]\{P\}+2\{P\}^T[B]\{P\}.BI_1(1,1)\right)$$
$$-\left(2\{P\}^T[BI_2]\{P\}.B(1,1)+4\{P\}_1^T[BD_1]\{P\}.\{P\}_2^T[BI_2]\{P\}\right.$$
$$\left.+4\{P\}_1^T[BI_2]\{P\}.\{P\}_2^T[B]\{P\}+2\{P\}^T[B]\{P\}.BI_2(1,1)\right)$$
$$+\left(2\{P\}^T[BI_3]\{P\}.BD_2(1,1)+4\{P\}_1^T[BD_2]\{P\}.\{P\}_2^T[BI_3]\{P\}\right.$$
$$\left.+4\{P\}_1^T[BI_3]\{P\}.\{P\}_2^T[BD_2]\{P\}+2\{P\}^T[B]\{P\}.BI_3(1,1)\right)$$
$$-\left(2\{P\}^T[BI_4]\{P\}.BD_3(1,1)+4\{P\}_1^T[BD_3]\{P\}.\{P\}_2^T[BI_4]\{P\}\right.$$
$$\left.+4\{P\}_1^T[BI_4]\{P\}.\{P\}_2^T[BD_3]\{P\}+2\{P\}^T[B]\{P\}.BI_4(1,1)\right)$$
$$+\left(2\{P\}^T[BI_5]\{P\}.BD_4(1,1)+4\{P\}_1^T[BD_4]\{P\}.\{P\}_2^T[BI_5]\{P\}\right.$$
$$\left.+4\{P\}_1^T[BI_5]\{P\}.\{P\}_2^T[BD_4]\{P\}+2\{P\}^T[B]\{P\}.BI_5(1,1)\right)$$
$$\left.-\frac{2}{2}-\left(2BI_1(1,1)\right)\right\}$$

Appendix B-2 (2D computer program list)

2Dprog first declares the variables and arrays to be used in the program. The coefficients in the arrays PG,DG,KG,KAG,KBG,KSG, FBG,FAG,FSG and FPG, are initially set to zero in the declaration.

2Dprog then calls the subroutines according to the flow diagram shown next. In the summary; the groups of subroutines indicated by the following functions;

1. INPDAT to BANDW The structural data is read and stored in appropriate arrays.
2. AANDTA [A] matrix is calculated
3. FORCE to KFPMAT The element forces are calculated and printed by cycling throught the elements.
4. UANDTU to KBGKAT The global stiffness matrices for the bending and axial are calculated.
5. SPRING to KSGMAT The global stiffness matrix of the spring is calculated by cycling through the boundires.
6. PGKMAR The global forces stiffness matrix.
7. SOLVE The equilibrium equations are solved and the dispacement soluton is printed.
8. OUTSO1 The coordinates of the final deformed shap are printed.



Flow-diagram of 2D-prog.

```
C=============================================================
C                      PROGRAM 2DIMEN.
C
C    THIS PROGRAM IS  DEVELOPED TO CARRY OUT  TWO-DIMENTIONAL
C            ANALYSIS OF  CURVES BY USING  THE FINITE
C    ELEMENT NON-LINEAR TECHNIQUE AND CUBIC B-SPLINE FUNCTION
C=============================================================
C
C      ------------------------------------------------------
C     |            LIST OF MAIN VARIABLES AND ARRAYS         |
C     |           ...................................        |
C     |                                                      |
C     |  NG     = NUMBER OF ELEMENT GROUPS                   |
C     |  NE     = NUMBER OF ELEMENT                          |
C     |  ROWN   = NUMBER OF ROWS IN THE G-STIFF. MATRIX      |
C     |  COLN   = NUMBER OF COLUMNS IN THE G-STIFF. MATRIX   |
C     |  NOCP   = NUMBER OF CONTROL POINTS                   |
C     |  NCPS   = NUMBER OF NODES PER SPLINE                 |
C     |  NDFP   = NUMBER OF DEGREES OF FREEDOM PER POINT     |
C     |  NFNO   = NUMBER OF EXTERNAL FORCES                  |
C     |  NRNO   = NUMBER OF RESTRAIN NODES                   |
C     |  DOF    = NUMBER OF TOTAL DEGREES OF FREEDOM         |
C     |  CORD   = CO-ORDINATES OF THE CONTROL POINTS         |
C     |  MCON   = SPLINE CONNECTIVITY                        |
C     |  PROP   = MATERIAL GROUP & CROSS SECTION PROPERTIES  |
C     |  PFR    = EXTERNAL FORCE PROPERTIES                  |
C     |  PFRC   = EXTERNAL FORCE POSITION                    |
C     |  SPR    = SPRNG PROPERTIES                           |
C     |  SPRE   = SPRNG POSITION                             |
C     |  KBE    = ELEMENT BENDING STIFF.                     |
C     |  KAE    = ELEMENT AXIAL STIFF.                       |
C     |  KS     = ELEMENT SPRING STIFF.                      |
C     |  FBE    = ELEMENT BENDING FORCE                      |
C     |  FAE    = ELEMENT AXIAL FORCE                        |
C     |  FS     = ELEMENT SPRING FORCE                       |
C     |  FPE    = ELEMENT APPLIED LOAD                       |
C     |  KBG    = GLOBAL BENDING STIFF.                      |
C     |  KAG    = GLOBAL AXIAL STIFF.                        |
C     |  KSG    = GLOBAL SPRING STIFF.                       |
C     |  FBG    = GLOBAL BENDING FORCE                       |
C     |  FAG    = GLOBAL AXIAL FORCE                         |
C     |  FPG    = GLOBAL EXTERNAL FORCE                      |
C     |  FSG    = GLOBAL SPRING FORCE                        |
C     |  KG     = ELASTIC LINEAR STIFFNESS                   |
C     |  PG     = GLOBAL FORCE MATRIX                        |
C     |  DG     = GLOBAL DISPLACEMENT MATRIX                 |
C     |  ALFA   = AXIAL STRAIN                               |
C     |  BETA   = BENDING  STRAIN                            |
C     |  LCNO   = NO. OF CASE OF LOADING                     |
C     |  IT     = ITERATION NUMBER.                          |
C     |  ACCU   = THE ITERATION ACCUARCY                     |
C     |  INCR   = THE LOAD INCREMENTS                        |
C
C=============================================================
      INTEGER NG,IE,NE,NR,NOCP,NDFP,DOF,JU,NFNO,NRNO
     *       ,NF,NCPS,ROWN,COLN,LCNO
C
```

```
      DOUBLE PRECISION  CORD(20,2),PROP(20,4),SPR(20,3)
     * ,FBE(8),FAE(8),A(4,4),TA(4,4),C1(4,4),C2(4,4),B(4,4)
     *   ,KBG(20,20),KAG(20,20),KSG(20,20),KG(20,20)
     *   ,U(1,4),U1(1,4),U2(1,4),TU1(4,1),TU2(4,1),INCR(4)
     *   ,EI(4,4),FI(4,4),GI(4,4),HI(4,4),SI(4,4),ACU(20)
     *   ,FBG(20),FAG(20),FPG(20),FSG(20),PG(20),DG(20)
     *   ,FBM(8),FAM(8),FPE(8),FS(8),WKA(8),PFR(20,2),X(20)
     *   ,IBB(5,8,8),BBD(5,8,8),KS(8,8),PT(20),DELTA(20)
     *   ,KBE(8,8),KAE(8,8),KBM(8,8),KAM(8,8),DH(8,8)
     *   ,BCPD(5,4,4),BI(5,4,4),D(4,4),DGX(20),DGY(20),PP(4)
     *   ,UA,ALFA,BETA,ACCU,PGG(20),KGG(20,20),PTT(4)
      INTEGER LN,IR(4),MCON(30,4),SPRE(20,2),PFRC(20,2)
C=============================================================
      CALL INPDAT(NG,NE,NCPS,NOCP,NDFP,DOF,CORD,X,MCON
     *     ,PROP,SPRE,SPR,NFNO,NRNO,ALFA,BETA,EI,FI,GI,HI,SI
     *     ,PFR,PFRC,LCNO,INCR)
      CALL OUPDAT(NG,NE,NCPS,NOCP,NDFP,DOF,CORD,X,MCON
     *     ,PROP,SPRE,SPR,NFNO,NRNO,PFR,PFRC,LCNO,INCR)
      CALL BANDW(NDFP,NOCP,NCPS,ROWN,COLN)
      CALL AANDTA(A,TA)
C
      DO 230 NF=1,NFNO
      PP(NF)=PFR(NF,2)
230   CONTINUE
      ITT=0
      DO 4848 I=1,ROWN
      PG(I)=0.0
      PGG(I)=0.0
      DO 4848 J=1,COLN
      KG(I,J)=0.0
      KGG(I,J)=0.0
4848  CONTINUE
C
      DO 900 LN=1,LCNO
      ITT=ITT+1
      WRITE(6,901)ITT
901   FORMAT(1H,'LOAD CASE NO. ',I3)
C
      DO 30 NF=1,NFNO
      CALL FORCE(PP,LN,NFNO,NF,PFR,PFRC,FPE,A,ACU,LCNO,INCR)
      CALL KFPMAT(MCON,NFNO,NF,ROWN,FPE,FPG,PFR,PFRC,IR)
      IF(ABS(ACU(NF)).LE.ABS(ACU(1)))  ACCU=ABS(ACU(NF))
      IF(ACU(NF).LE.ACU(1))  ACCU=ACU(NF)
30    CONTINUE
      DO 601 K=1,ROWN
      PGG(K)=FPG(K)
601   CONTINUE
C
      IT=0
400   IT=IT+1
      WRITE(6,45)IT
45    FORMAT(1H,'ITERATION NO.  ',I5)
C
      DO 10 IE=1,NE
      DO 11 JU=1,2
      CALL UANDTU(JU,UA,U,U1,U2,TU1,TU2)
      CALL BMAT(IE,JU,A,TA,UA,U1,TU1,C1,B,EI,FI,GI,HI,SI,IBB,BBD,DH)
```

```
      CALL FBEMAT(IE,JU,UA,FBE,BETA,X,ROWN,FBM,DH,PROP)
      CALL KBEMAT(IE,JU,UA,KBE,BETA,X,ROWN,KBM,DH,PROP)
      CALL FAEMAT(IE,JU,X,FAE,ALFA,ROWN,PROP,FAM,IBB,BBD,BI,BD)
      CALL KAEMAT(IE,JU,X,KAE,ALFA,ROWN,PROP,KAM,IBB,BBD)
11    CONTINUE
      CALL FBGFAG(IE,NE,FBM,FAM,FBG,FAG,ROWN,MCON)
      CALL KBGKAG(IE,NE,KBM,KAM,KBG,KAG,ROWN,COLN,MCON)
10    CONTINUE
      DO 20 NR=1,NRNO
      CALL SPRNGF(NR,MCON,NRNO,A,SPRE,SPR,FS,X,ROWN,IR)
      CALL SPRNGK(NR,MCON,NRNO,A,SPRE,SPR,KS,X,ROWN,IR)
      CALL FSGMAT(NR,MCON,NRNO,ROWN,FS,FSG,SPRE,IR)
      CALL KSGMAT(NR,MCON,NRNO,ROWN,COLN,KS,KSG,SPRE,IR)
20    CONTINUE
      CALL PGKGMT(FBG,FAG,FSG,FPG,PG,KBG,KAG,KSG,KG,ROWN
     *           ,COLN,KGG,PGG,PT,DELTA)
      WRITE(6,404)
404   FORMAT(1H,'KG(20,20)')
      WRITE(6,405)((KG(I,J),J=1,COLN),I=1,ROWN)
405   FORMAT(8E11.4)
      WRITE(6,303)
303   FORMAT(1H,'PG')
      WRITE(6,304)(PG(I),I=1,ROWN)
304   FORMAT(E14.5)
      WRITE(6,363)
363   FORMAT(1H,'DG')
      WRITE(6,364)(DG(I),I=1,ROWN)
364   FORMAT(E14.5)
      CALL SOLVE (ROWN,COLN,KG,PG,WKA,DG)
      DO 50 I=1,ROWN
      X(I)=X(I)+DG(I)
      WRITE(6,4050)DG(I)
4050  FORMAT(F17.6)
50    CONTINUE
      DO 300 I=1,ROWN
      IF(DELTA(I).GT.ABS(ACCU))GOTO 400
300   CONTINUE
      CALL OUTSO1(NE,MCON,NOCP,ROWN,DG,DOF,A,DGX,DGY,X)
900   CONTINUE
      CLOSE(6)
      CLOSE(5)
C
      STOP
      END
C=============================================================
C     SUBROUTINE IN PUT DATA
C=============================================================
      SUBROUTINE INPDAT(NG,NE,NCPS,NOCP,NDFP,DOF,CORD,X,MCON
     *     ,PROP,SPRE,SPR,NFNO,NRNO,ALFA,BETA,EI,FI,GI,HI,SI
     *     ,PFR,PFRC,LCNO,INCR)
C
      INTEGER  NG,NE,NOCP,NDFP,DOF,NFNO,NRNO,NCPS,LCNO
      DOUBLE PRECISION   CORD(20,2),PROP(20,4),SPR(20,3)
     *        ,EI(4,4),FI(4,4),GI(4,4),HI(4,4),SI(4,4)
     *        ,X(20),PFR(20,2),ALFA,BETA,INCR(4)
      INTEGER  MCON(30,4),SPRE(20,2),PFRC(20,2)
C
```

```
      READ(5,*)NOCP,NE,NG,NDFP,DOF,NFNO,NRNO,NCPS
      READ(5,*) (M,(CORD(M,N),N=1,2),I=1,NOCP)
      DO 39 JJ=1,NOCP
      X(2*JJ-1)=CORD(JJ,1)
      X(2*JJ)  =CORD(JJ,2)
39    CONTINUE
C
      READ(5,*) (M,(MCON(M,N),N=1,4),I=1,NE)
      DO 800 II=1,NG
      READ(5,*) M,(PROP(II,J),J=1,4)
      ALFA=PROP(II,1)*PROP(II,3)*(PROP(II,4))
      BETA=PROP(II,1)*PROP(II,2)/((PROP(II,4))**3.0)
800   CONTINUE
      READ(5,*)LCNO,(INCR(K),K=1,3)
      READ(5,*) (M,(PFRC(M,N),N=1,2),(PFR(M,N),N=1,2),M=1,NFNO)
      READ(5,*) (M,(SPRE(M,N),N=1,2),(SPR(M,N),N=1,3),M=1,NRNO)
      READ(5,*) (M,(EI(M,N),N=1,4),I=1,NCPS)
      READ(5,*) (M,(FI(M,N),N=1,4),I=1,NCPS)
      READ(5,*) (M,(GI(M,N),N=1,4),I=1,NCPS)
      READ(5,*) (M,(HI(M,N),N=1,4),I=1,NCPS)
      READ(5,*) (M,(SI(M,N),N=1,4),I=1,NCPS)
C
      RETURN
      END
C=============================================================
C     SUBROUTINE OUT PUT DATA
C=============================================================
      SUBROUTINE OUPDAT(NG,NE,NCPS,NOCP,NDFP,DOF,CORD,X,MCON
     *          ,PROP,SPRE,SPR,NFNO,NRNO,PFR,PFRC,LCNO,INCR)
C
      INTEGER    NG,NE,NOCP,NDFP,DOF,NFNO,NRNO,NCPS,LCNO
      DOUBLE PRECISION   CORD(20,2),PROP(20,4),SPR(20,3)
     *                  ,INCR(4),X(20),PFR(20,2)
      INTEGER  MCON(30,4),SPRE(20,2),PFRC(20,2)
C
      WRITE(6,700)
700   FORMAT(//22X,'THE DATA'/22X,'========')
      WRITE(6,710)NG,NE,NOCP,NDFP,DOF,NFNO,NRNO,NCPS
710   FORMAT(1H,'DATA SUPPLIED',//1H,'NUMBER OF ELE.GROUP',I5,
     *          /1H,'NUMBER OF ELEMENT       ',I5,
     *          /1H,'NUMBER OF CONTROL POINT ',I5,
     *          /1H,'NUM. OF D.O.F PER N.C.P ',I5,
     *          /1H,'TOTAL NUM. OF D.O.F     ',I5,
     *          /1H,'NUMBER OF NODEL FORCES  ',I5,
     *          /1H,'NUMBER OF SPRING        ',I5,
     *          /1H,'NO. OF C.P. PER SPLINE  ',I5)
      WRITE(6,720)
720   FORMAT(//1H,'NODES COORDINATE',/1H,8X,'NODE',9X,
     *         'X(1)',10X,'X(2)')
      WRITE(6,730) (M,(CORD(M,N),N=1,2),M=1,NOCP)
730   FORMAT(1H,I10,2F15.5)
      WRITE(6,740)
740   FORMAT(//1H,'MEMBER CONNECTIVITY',/1H,2X,'MEMBER',
     *   6X,'NODE 1',6X,'NODE 2',6X,'NODE 3',6X,'NODE 4')
      WRITE(6,750) (M,(MCON(M,N),N=1,4),M=1,NE)
750   FORMAT(1H,I6,4I12)
      WRITE(6,760)
```

```
229   760   FORMAT(//1H,'MEMBER PROPERTIES',/1H,1X,'MEMBER',
230        *6X,'E(N/ME)',7X,'I(M4)',7X,'AREA(M2)',7X,'LENGTH(M)')
231         WRITE(6,770)(M,(PROP(M,N),N=1,4),M=1,NG)
232   770   FORMAT(1H,I5,4E14.4)
233         WRITE(6,776)
234   776   FORMAT(//1H,'APPLIED LOADS',/1H,4X,'LOAD NO.',4X,
235        *'ELEM.NO.',4X,'DIRECTION',8X,'(U)',5X,'LOAD VALUE')
236         WRITE(6,777)(M,(PFRC(M,N),N=1,2),(PFR(M,N),N=1,2),M=1,NFNO)
237   777   FORMAT(1H,I10,2I12,F14.3,E14.5)
238         WRITE(6,780)
239   780   FORMAT(//1H,'SPRING PROPERTIES',/1H,4X,'SPRING NO.',3X,
240        *'ELEM.NO.',3X,'DIRECTION',8X,'(U)',9X,'(S)',10X,'CONST.')
241         WRITE(6,790)(M,(SPRE(M,N),N=1,2),(SPR(M,N),N=1,3),M=1,NRNO)
242   790   FORMAT(1H,I10,2I12,F14.3,E14.5,F14.3)
243   C
244         RETURN
245         END
246   C=============================================================
247   C     SUB. TO CALCULAT THE BAND WIDTH OF THE STIFF. MATRICX
248   C=============================================================
249         SUBROUTINE BANDW(NDFP,NOCP,NCPS,ROWN,COLN)
250   C
251         INTEGER   NDFP,NCPS,COLN,ROWN,NOCP
252   C
253         COLN=NDFP*NOCP
254         ROWN=NDFP*NOCP
255   C
256         RETURN
257         END
258   C=============================================================
259   C     SUB. FOR CALCULATE THE A MATRIX
260   C=============================================================
261         SUBROUTINE AANDTA(A,TA)
262   C
263         DOUBLE PRECISION  A(4,4),TA(4,4),DD
264   C
265         DD=1.0/6.0
266         A(1,1)= 1*DD
267         A(1,2)= 4*DD
68         A(1,3)= 1*DD
269         A(2,1)=-3*DD
270         A(2,3)= 3*DD
271         A(3,1)= 3*DD
272         A(3,2)=-6*DD
273         A(3,3)= 3*DD
274         A(4,1)=-1*DD
275         A(4,2)= 3*DD
276         A(4,3)=-3*DD
277         A(4,4)= 1*DD
278   C
279         DO 10 I=1,4
280         DO 10 J=1,4
281         TA(I,J)=A(J,I)
282   10    CONTINUE
283   C
284         RETURN
285         END
```

```
C=============================================================
C     SUBROUTINE UANDA TO CALCU. [U] MATRICES.
C=============================================================
      SUBROUTINE UANDTU(JU,UA,U,U1,U2,TU1,TU2)
C
      INTEGER  JU
      DOUBLE PRECISION UA,U(1,4),U1(1,4),U2(1,4),TU1(4,1),TU2(4,1)
C
      IF(JU.EQ.1) UA=1.0
      IF(JU.EQ.2) UA=0.0
      U(1,1)=1.0
      U(1,2)=UA
      U(1,3)=UA**2
      U(1,4)=UA**3
C
      U1(1,2)=1
      U1(1,3)=2*UA
      U1(1,4)=3*(UA**2)
C
      TU1(2,1)=U1(1,2)
      TU1(3,1)=U1(1,3)
      TU1(4,1)=U1(1,4)
C
      U2(1,3)=2.0
      U2(1,4)=6*UA
C
      TU2(3,1)=U2(1,3)
      TU2(4,1)=U2(1,4)
C
      RETURN
      END
C=============================================================
C     SUBROUTINE BMAT TO CALCU. [B] MATRIX
C=============================================================
      SUBROUTINE BMAT(IE,JU,A,TA,UA,U1,TU1,C1,B,EI,FI,GI,HI
     *                ,SI,IBB,BBD,DH)
C
      INTEGER JU
C
      DOUBLE PRECISION A(4,4),TA(4,4),C1(4,4),B(4,4),B1(4,4)
     *    ,EI(4,4),FI(4,4),GI(4,4),HI(4,4),SI(4,4),D(4,4)
     *    ,UA,U1(1,4),TU1(4,1),CP(4,4),BIC(5,4,4),DH(8,8)
     *    ,IBB(5,8,8),BBD(5,8,8),BI(5,4,4),BD(5,4,4)
C
      DO 20 L=1,4
      DO 20 K=1,4
      C1(L,K)=TU1(L,1)*U1(1,K)
20    CONTINUE
C
C     MATRIX [B]=[A]'[U']'[U'][A]
C     ===========================
      DO 31 I=1,4
      DO 31 J=1,4
      B1(I,J)=0.0
      DO 31 M=1,4
      B1(I,J)=B1(I,J)+TA(I,M)*C1(M,J)
31    CONTINUE
```

```
343   C
344         DO 32 II=1,4
345         DO 32 JJ=1,4
346         B(II,JJ)=0.0
347         DO 32 MM=1,4
348         B(II,JJ)=B(II,JJ)+B1(II,MM)*A(MM,JJ)
349   32    CONTINUE
350   C
351   C     CALCULATION OF THE 1st,2nd,....,5th INTEGRATION OF [C]
352   C     ======================================================
353         U11=UA
354         U22=UA**2
355         U33=UA**3
356         U44=UA**4
357         U55=UA**5
358         U66=UA**6
359         U77=UA**7
360         U88=UA**8
361         U99=UA**9
362   C
363         D(1,1)= 6 *U11-6*U22+2*U33
364         D(1,2)=-12*U11+15*U22-6*U33
365         D(1,3)= 6 *U11-12*U22+6*U33
366         D(1,4)= 3 *U22-2*U33
367         D(2,2)= 24*U11-36*U22+18*U33
368         D(2,3)=-12*U11+27*U22-18*U33
369         D(2,4)=-6 *U22+6*U33
370         D(3,3)= 6 *U11-18*U22+18*U33
371         D(3,4)= 3 *U22-6*U33
372         D(4,4)= 2 *U33
373         DO 76 I=1,4
374         DO 76 J=1,4
375         D(I,J)=D(I,J)/6.0
376         IF(I.GT.J) D(I,J)=D(J,I)
377   76    CONTINUE
378         DO 71 L=1,4
379         LL=2*L
380         DO 71 M=1,4
381         MM=2*M
382         DH(LL,MM)=D(L,M)
383         DH(LL-1,MM-1)=D(L,M)
384   71    CONTINUE
385   C
386         DO 48 K=1,4
387         DO 48 L=1,4
388         BI(1,K,L)=(U11*EI(K,L) + U22*FI(K,L)/2.0 + U33*GI(K,L)
389        *     /3.0 + U44*HI(K,L)/4.0 + U55*SI(K,L)/5.0)/4.0
390         BI(2,K,L)=(U22*EI(K,L)/2.0 + U33*FI(K,L)/6.0  +U44*GI(K,L)
391        *    /12.0 + U55*HI(K,L)/20.0+ U66*SI(K,L)/30.0)/4.0
392         BI(3,K,L)=(U33*EI(K,L)/6.0 + U44*FI(K,L)/24.0 +U55*GI(K,L)
393        *    /60.0 + U66*HI(K,L)/120.0+U77*SI(K,L)/210.0)/4.0
394         BI(4,K,L)=(U44*EI(K,L)/24.0 +U55*FI(K,L)/120.0+U66*GI(K,L)
395        *    /360.0+ U77*HI(K,L)/840.0+U88*SI(K,L)/1680.0)/4.0
396         BI(5,K,L)=(U55*EI(K,L)/120.0+U66*FI(K,L)/720.0+U77*GI(K,L)
397        *    /2520 + U88*HI(K,L)/6720.0+U99*SI(K,L)/15120.0)/4.0
398   C
399   C     CALCULATION OF THE 1st,2nd,3rd,4th DIFFERETIATION OF [C]
```

```
C     ============================================================
      BD(1,K,L)=(0.25)*(EI(K,L)+U11*FI(K,L)+U22*GI(K,L)+U33*
     *                  HI(K,L)+U44*SI(K,L))
      BD(2,K,L)=(0.25)*(FI(K,L)+2.0*U11*GI(K,L)+3.0*U22*HI(K,L)
     *                 +4.0*U33*SI(K,L))
      BD(3,K,L)=(0.25)*(2.0*GI(K,L)+6.0*U11*HI(K,L)+12.0*U22*SI(K,L))
      BD(4,K,L)=(0.25)*(6.0*HI(K,L)+24.0*U11*SI(K,L))
      BD(5,K,L)=(0.25)*(24.0*SI(K,L))
48    CONTINUE
C
C     TRANSFORMATION BI(4,4) TO IBB(8,8) & BD(4,4) TO BBD(8,8)
C     ============================================================
      YY=0
      DO 90 L=1,7,2
      KY=0.0
      DO 9 K=1,4
      DO 901 KK=1,5
      IBB(KK,L,K+KY)=BI(KK,L-YY,K)
      BBD(KK,L,K+KY)=BD(KK,L-YY,K)
      IBB(KK,L+1,K+KY+1)=BI(KK,L-YY,K)
      BBD(KK,L+1,K+KY+1)=BD(KK,L-YY,K)
901   CONTINUE
      KY=KY+1
9     CONTINUE
      YY=YY+1
90    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUB. TO CALCULAT BENDING FORCE  MATRICES.
C=============================================================
      SUBROUTINE FBEMAT(IE,JU,UA,FBE,BETA,X,ROWN,FBM,DH,PROP)
C
      INTEGER     ROWN,JU,IE
      DOUBLE PRECISION  BETA,UA,X(20),DH(8,8),LEN,PROP(20,4)
     *                 ,FBE(8),FBEU(8),FBM(8)
C
      LEN=PROP(1,4)
      JD=2*IE-2
      DO 25 I=1,8
      FBE(I)=0.0
      DO 25 J=1,8
      JJ=J+JD
      FBE(I)=FBE(I)+DH(I,J)*X(JJ)
25    CONTINUE
C
      IF(JU.EQ.1) THEN
      DO 250 M=1,8
      FBEU(M)=FBE(M)
250   CONTINUE
      ENDIF
C
      IF(JU.EQ.2) THEN
      DO 33 M=1,8
      FBM(M)=BETA*(FBEU(M)-FBE(M))
33    CONTINUE
```

```
      ENDIF
C
      RETURN
      END
C=======================================================
C     SUB. TO CALCULAT BENDING  STIFFENS MATRICES.
C=======================================================
      SUBROUTINE KBEMAT(IE,JU,UA,KBE,BETA,X,ROWN,KBM,DH,PROP)
C
      INTEGER     ROWN,JU,IE
      DOUBLE PRECISION   BETA,UA,X(20),DH(8,8),PROP(20,4),LEN
     *                  ,KBE(8,8),KBM(8,8),KBEU(8,8)
C
      LEN=PROP(1,4)
      DO 25 I=1,8
      DO 25 J=1,8
      KBE(I,J)=DH(I,J)
25    CONTINUE
C
      IF(JU.EQ.1) THEN
      DO 250 M=1,8
      DO 250 N=1,8
      KBEU(M,N)=KBE(M,N)
250   CONTINUE
      ENDIF
C
      IF(JU.EQ.2) THEN
      DO 33 M=1,8
      DO 33 N=1,8
      KBM(M,N)=BETA*(KBEU(M,N)-KBE(M,N))
33    CONTINUE
      ENDIF
C
      RETURN
      END
C=======================================================
C     SUB. TO CALCULAT AXIAL FORCE  MATRICES.
C=======================================================
      SUBROUTINE FAEMAT(IE,JU,X,FAE,ALFA,ROWN,PROP,FAM,IBB,BBD
     *                        ,BI,BD)
C
      INTEGER  ROWN,JU,IE
C
      DOUBLE PRECISION X(20),FAE2(8),FAE1(8),PROP(20,4)
     *    ,FAE(8),FAM(8),FAEU(8),BC1(5,8),BC2(5,8),BI(5,4,4)
     *    ,IBB(5,8,8),BBD(5,8,8),PP1(5),PP2(5),BD(5,4,4)
     *    ,ALFA,LEN
C
      LEN=PROP(1,4)
      JD=2*IE-2
C
      DO 101 JB=1,5
      PP1(JB)=0.0
      PP2(JB)=0.0
      DO 101 I=1,8
      BC1(JB,I)=0.0
      BC2(JB,I)=0.0
```

```
514   101   CONTINUE
515   C
516         DO 103 JB=1,5
517         DO 33 I=1,8
518         DO 13 J=1,8
519         BC1(JB,I)=BC1(JB,I)+BBD(JB,I,J)*X(J+JD)
520         BC2(JB,I)=BC2(JB,I)+IBB(JB,I,J)*X(J+JD)
521   13    CONTINUE
522         PP1(JB)   =PP1(JB)+X(I+JD)*BC2(JB,I)
523         PP2(JB)   =PP2(JB)+X(I+JD)*BC1(JB,I)
524   33    CONTINUE
525   103   CONTINUE
526         DO 30 I=1,8
527         FAE1(I)=0.0
528         IK=-1
529         DO 301 M=1,5
530         IK=-IK
531         FAE1(I)=FAE1(I)+IK*(2*PP1(M)*BC1(M,I)+2*PP2(M)*BC2(M,I))
532   301   CONTINUE
533   30    CONTINUE
534   C
535         DO 92 I=1,8
536         FAE2(I)=2*BC2(1,I)
537   92    CONTINUE
538   C
539         DO 130 I=1,8
540         FAE1(I)=FAE1(I)/(LEN**4)
541         FAE2(I)=FAE2(I)*(2/(LEN**2))
542         FAE(I)=(ALFA/8)*(FAE1(I)-FAE2(I))
543   130   CONTINUE
544   C
545         IF(JU.EQ.1) THEN
546         DO 505 I=1,8
547         FAEU(I)=FAE(I)
548   505   CONTINUE
549         ENDIF
550   C
551         IF(JU.EQ.2) THEN
552         DO 50 I=1,8
553         FAM(I)=FAEU(I)-FAE(I)
554   50    CONTINUE
555         ENDIF
556   C
557         RETURN
558         END
559   C=============================================================
560   C     SUB. TO CALCULAT AXIAL  STIFFNES MATRICS.
561   C=============================================================
562         SUBROUTINE KAEMAT(IE,JU,X,KAE,ALFA,ROWN,PROP,KAM,IBB,BBD)
563   C
564         INTEGER  ROWN,JU,IE
565   C
566         DOUBLE PRECISION X(20),KAE2(8,8),KAE1(8,8),PROP(20,4)
567        *     ,KAE(8,8),KAM(8,8),KAEU(8,8),BD(5,4,4)
568        *     ,PP1(5),PP2(5),BC1(5,8),BC2(5,8)
569        *     ,ALFA,LEN,IBB(5,8,8),BBD(5,8,8)
570   C
```

```
      LEN=PROP(1,4)
      JD=2*IE-2
C
      DO 101 JB=1,5
      PP1(JB)=0.0
      PP2(JB)=0.0
      DO 101 I=1,8
      BC1(JB,I)=0.0
      BC2(JB,I)=0.0
101   CONTINUE
C
      DO 103 JB=1,5
      DO 33 I=1,8
      DO 13 J=1,8
      BC1(JB,I)=BC1(JB,I)+BBD(JB,I,J)*X(J+JD)
      BC2(JB,I)=BC2(JB,I)+IBB(JB,I,J)*X(J+JD)
13    CONTINUE
      PP1(JB)  =PP1(JB)+X(I+JD)*BC2(JB,I)
      PP2(JB)  =PP2(JB)+X(I+JD)*BC1(JB,I)
33    CONTINUE
103   CONTINUE
C
      DO 82 I=1,8
      DO 82 J=1,8
      KAE1(I,J)=0.0
      IK=-1
      DO 83 JB=1,5
      IK=-IK
      KAE1(I,J)=KAE1(I,J)+IK*(2*PP1(JB)*BBD(JB,I,J)
     *                        +4*BC1(JB,I)*BC2(JB,J)
     *                        +4*BC2(JB,I)*BC1(JB,J)
     *                        +2*PP2(JB)*IBB(JB,I,J))
83    CONTINUE
82    CONTINUE
C
      DO 87 I=1,8
      DO 87 J=1,8
      KAE2(I,J)=2*IBB(1,I,J)
87    CONTINUE
C
      DO 130 I=1,8
      DO 130 J=1,8
      KAE1(I,J)=KAE1(I,J)/(LEN**4)
      KAE2(I,J)=KAE2(I,J)*(2/(LEN**2))
      KAE(I,J)=(ALFA/8)*(KAE1(I,J)-KAE2(I,J))
130   CONTINUE
C
      IF(JU.EQ.1) THEN
      DO 505 I=1,8
      DO 505 J=1,8
      KAEU(I,J)=KAE(I,J)
505   CONTINUE
      ENDIF
C
      IF(JU.EQ.2) THEN
      DO 50 I=1,8
      DO 50 J=1,8
```

```
628         KAM(I,J)=KAEU(I,J)-KAE(I,J)
629    50   CONTINUE
630         ENDIF
631    C
632         RETURN
633         END
634    C=============================================================
635    C     SUB. TO CALCULAT BENDING & AXIAL GLOBAL FORCE
636    C=============================================================
637         SUBROUTINE FBGFAG(IE,NE,FBM,FAM,FBG,FAG,ROWN,MCON)
638    C
639         INTEGER  IE,NE,ROWN,MCON(30,4),H(4),V(4),P(4)
640    C
641         DOUBLE PRECISION  FBM(8),FAM(8),FBG(20),FAG(20)
642    C
643         IF(IE.NE.1) GO TO 40
644         DO 50 L=1,ROWN
645         FBG(L)=0.0
646    50   FAG(L)=0.0
647    40   CONTINUE
648    C
649         DO 10 K=1,4
650         H(K)=MCON(IE,K)
651         V(K)=2*(H(K)-H(1))
652    10   CONTINUE
653    C
654         DO 60 M=1,2
655         DO 20 I=1,4
656         P(I)=(2*H(I)-1)+M-1
657    20   CONTINUE
658    C
659         DO 30 L=1,4
660         FBG(P(L))=FBG(P(L))+FBM(M+V(L))
661         FAG(P(L))=FAG(P(L))+FAM(M+V(L))
662    30   CONTINUE
663    60   CONTINUE
664    C
665         RETURN
666         END
667    C=============================================================
668    C     SUB. TO CALCULAT THE GLOBAL BENDING & AXIAL STIFF.MAT.
669    C=============================================================
670         SUBROUTINE KBGKAG(IE,NE,KBM,KAM,KBG,KAG,ROWN,COLN,MCON)
671    C
672         INTEGER  IE,NE,ROWN,COLN,MCON(30,4),H(4),V(4),P(4),PP(4)
673    C
674         DOUBLE PRECISION KBG(20,20),KAG(20,20),KBM(8,8),KAM(8,8)
675    C
676         IF(IE.NE.1) GO TO 20
677         DO 30 M=1,ROWN
678         DO 30 N=1,COLN
679         KBG(M,N)=0.0
680    30   KAG(M,N)=0.0
681    20   CONTINUE
682    C
683         DO 10 K=1,4
684         H(K)=MCON(IE,K)
```

```
685         V(K)=2*(H(K)-H(1))
686   10    CONTINUE
687   C
688         DO 35 L=1,4
689         P(L)=2*H(L)-1
690   35    CONTINUE
691         DO 31 I=1,4
692         DO 31 J=1,4
693         DO 40 M=1,2
694         DO 40 N=1,2
695         II=P(I)+M-1
696         JJ=P(J)+N-1
697         KBG(II,JJ)=KBG(II,JJ)+KBM(V(I)+M,V(J)+N)
698         KAG(II,JJ)=KAG(II,JJ)+KAM(V(I)+M,V(J)+N)
699   40    CONTINUE
700   31    CONTINUE
701   C
702         RETURN
703         END
704   C=============================================================
705   C     SUB. FOR CALCULATING THE EQUIVLANT FORCE ON THE C.P.
706   C=============================================================
707         SUBROUTINE FORCE(PP,LN,NFNO,NF,PFR,PFRC,FPE,A,ACU,LCNO,INCR)
708   C
709         INTEGER    IE,NFNO,NF,ROWN,LCNO,LN,PFRC(20,2),IR(4)
710   C
711         DOUBLE PRECISION   INCR(4),A(4,4),PFR(20,2),FPE(8)
712        *           ,ACU(20),U(1,4),C(1,4),UA(4),PP(4),DELP(4)
713   C
714         DO 10 I=1,8
715         FPE(I)=0.0
716   10    CONTINUE
717   C
718         DELP(NF)=INCR(NF)
719         IF(LN.EQ.1) DELP(NF)=0.0
720         PP(NF) =PP(NF)+DELP(NF)
721         UA(NF) =PFR(NF,1)
722         ACU(NF)=PP(NF)/10000.0
723         U(1,1)=1.0
724         IF(NF.EQ.3) ACU(3)=ACU(2)
725         U(1,2)=UA(NF)
726         U(1,3)=UA(NF)**2
727         U(1,4)=UA(NF)**3
728         DO 300 L=1,4
729         C(1,L)=0.00
730         DO 30 K=1,4
731         C(1,L)=C(1,L)+U(1,K)*A(K,L)
732   30    CONTINUE
733   300   CONTINUE
734   C
735         DO 40 J=1,4
736         IF(PFRC(NF,2).EQ.2) GO TO 17
737         JS=2*J-1
738         GO TO 18
739   17    JS=2*J
740   18    FPE(JS)=FPE(JS)+PP(NF)*C(1,J)
741   40    CONTINUE
```

```
C
      RETURN
      END
C=============================================================
C     SUB. FOR CALCULATING THE GLOBAL FORCE & STIFF.LOAD MAT.
C=============================================================
      SUBROUTINE KFPMAT(MCON,NFNO,NF,ROWN,FPE,FPG,PFR,PFRC,IR)
C
      INTEGER  NFNO,NF,ROWN,MCON(30,4),PFRC(20,2),IR(4)
C
      DOUBLE PRECISION   FPE(8),FPG(20),PFR(20,2)
C
      IF(NF.NE.1) GO TO 75
      DO 70 I=1,ROWN
      FPG(I)=0.0
70    CONTINUE
C
75    MF=PFRC(NF,1)
      DO 35 J=1,4
      IR(J)=MCON(MF,J)
35    CONTINUE
C
      DO 45 N=1,4
      DO 45 M=1,2
      IQ=2*IR(N)+M-2
      JQ=2*N+M-2
      FPG(IQ)=FPG(IQ)+FPE(JQ)
45    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUBROUTINE SPRNG
C=============================================================
      SUBROUTINE SPRNGF(NR,MCON,NRNO,A,SPRE,SPR,FS,X,ROWN,IR)
C
      INTEGER  NR,NRNO,ROWN,MCON(30,4),SPRE(20,2),IR(4)
C
      DOUBLE PRECISION   SPR(20,3),A(4,4),C(1,4),U(1,4)
     *        ,X(20),DE(10),DEL(10),FS(8),UA,S,CONST
C
      DO 10 I=1,8
      FS(I)=0.0
10    CONTINUE
C
      UA   =SPR(NR,1)
      S    =SPR(NR,2)
      CONST=SPR(NR,3)
      U(1,1)=1
      U(1,2)=UA
      U(1,3)=UA**2
      U(1,4)=UA**3
      DO 30 L=1,4
      C(1,L)=0.0
      DO 30 K=1,4
      C(1,L)=C(1,L)+U(1,K)*A(K,L)
30    CONTINUE
```

```
C
      MR=SPRE(NR,1)
      DE(NR)=0.0
      DO 35 J=1,4
      IR(J)=MCON(MR,J)
      IF(SPRE(NR,2).EQ.2) GO TO 15
      JX=2*IR(J)-1
      GO TO 16
15    JX=2*IR(J)
16    DE(NR)=DE(NR)+C(1,J)*X(JX)
35    CONTINUE
      DEL(NR)=DE(NR)-CONST
C
      DO 40 J=1,4
      IF(SPRE(NR,2).EQ.2) GO TO 17
      JS=2*J-1
      GO TO 18
17    JS=2*J
18    FS(JS)=S*DEL(NR)*C(1,J)
40    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUB. SPRNGK FOR CALCULATING THE ELEMENT SPRING STIFF.M.
C=============================================================
      SUBROUTINE SPRNGK(NR,MCON,NRNO,A,SPRE,SPR,KS,X,ROWN,IR)
C
      INTEGER  NR,NRNO,ROWN,MCON(30,4),SPRE(20,2),IR(4)
C
      DOUBLE PRECISION SPR(20,3),A(4,4),C(1,4),U(1,4),KS(8,8)
     *                 ,X(20),DE(10),DEL(10),UA,S,CONST
C
      DO 10 I=1,8
      DO 10 J=1,8
      KS(I,J)=0.0
10    CONTINUE
C
      UA   =SPR(NR,1)
      S    =SPR(NR,2)
      CONST=SPR(NR,3)
      U(1,1)=1
      U(1,2)=UA
      U(1,3)=UA**2
      U(1,4)=UA**3
      DO 30 L=1,4
      C(1,L)=0.0
      DO 30 K=1,4
      C(1,L)=C(1,L)+U(1,K)*A(K,L)
30    CONTINUE
C
      DO 60 M=1,4
      DO 60 N=1,4
      IF(SPRE(NR,2).EQ.2) GO TO 36
      MM=2*M-1
      NN=2*N-1
      GO TO 37
```

```
856   36    MM=2*M
857         NN=2*N
858   37    KS(MM,NN)=S*C(1,N)*C(1,M)
859   60    CONTINUE
860   C
861         RETURN
862         END
863   C=============================================================
864   C     SUB. TO CALCULAT THE SPRING GOLABL FORCE  MATRIX
865   C=============================================================
866         SUBROUTINE FSGMAT(NR,MCON,NRNO,ROWN,FS,FSG,SPRE,IR)
867   C
868         INTEGER    NR,NRNO,ROWN,IR(4),SPRE(20,2),MCON(30,4)
869         DOUBLE PRECISION   FS(8),FSG(20)
870   C
871         IF(NR.NE.1) GO TO 75
872         DO 70 I=1,ROWN
873   70    FSG(I)=0.0
874   75    CONTINUE
875   C
876         MR=SPRE(NR,1)
877         DO 350 J=1,4
878         IR(J)=MCON(MR,J)
879   350   CONTINUE
880   C
881         DO 45 N=1,4
882         DO 45 M=1,2
883         IQ=2*IR(N)+M-2
884         JQ=2*N+M-2
885         FSG(IQ)=FSG(IQ)+FS(JQ)
886   45    CONTINUE
887   C
888         RETURN
889         END
890   C=============================================================
891   C     SUB. TO CALCULAT THE SPRING GOLABL  STIFF. MATRIX
892   C=============================================================
893         SUBROUTINE KSGMAT(NR,MCON,NRNO,ROWN,COLN,KS,KSG,SPRE,IR)
894   C
895         INTEGER    NR,NRNO,ROWN,COLN,H(4)
896   C
897         DOUBLE PRECISION   KS(8,8),KSG(20,20)
898         INTEGER  IR(4),V(4),PP(4),P(4),MCON(30,4),SPRE(20,2)
899   C
900         IF(NR.NE.1) GO TO 75
901         DO 70 I=1,ROWN
902         DO 70 J=1,COLN
903   70    KSG(I,J)=0.0
904   75    CONTINUE
905   C
906         MR=SPRE(NR,1)
907         DO 10 K=1,4
908         H(K)=MCON(MR,K)
909         V(K)=2*(H(K)-H(1))
910   10    CONTINUE
911   C
912         DO 35 L=1,4
```

```
      P(L)=2*H(L)-1
35    CONTINUE
C
      DO 31 I=1,4
      DO 31 J=1,4
      DO 50 M=1,2
      DO 50 N=1,2
      II=P(I)+M-1
      JJ=P(J)+N-1
      KSG(II,JJ)=KSG(II,JJ)+KS(V(I)+M,V(J)+N)
50    CONTINUE
31    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUB. TO CALCULAT THE TOTAL FORCE AND STIFFNES MATRIX
C=============================================================
      SUBROUTINE PGKGMT(FBG,FAG,FSG,FPG,PG,KBG,KAG,KSG,KG
     *                  ,ROWN,COLN,KGG,PGG,PT,DELTA)
C
      INTEGER ROWN,COLN
C
      DOUBLE PRECISION KBG(20,20),KAG(20,20),KSG(20,20)
     *     ,FBG(20),FAG(20),FPG(20),FSG(20),PG(20),PGG(20)
     *     ,DELTA(20),PT(20),KG(20,20),KGG(20,20)
C
      DO 10 I=1,ROWN
      PT(I)=FBG(I)+FAG(I)+FSG(I)
      PG(I)=PGG(I)-PT(I)
      DELTA(I)=PG(I)
C
      DO 20 J=1,COLN
      KG(I,J)=KBG(I,J)+KAG(I,J)+KSG(I,J)
      KGG(I,J)=KG(I,J)
20    CONTINUE
10    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUB. SOLVE BY USING GUASS-ELEMENATION
C=============================================================
      SUBROUTINE    SOLVE(NX,MX,KG,BY,WKA,DY)
C
      INTEGER       NX,MX
      DOUBLE PRECISION BY(20),WKA(20),DY(20),BBY(20),KG(20,20)
C
       M1=NX
       M2=M1-1
       DO 300 I=1,M2
       II=I+1
       DO 290 K=II,M1
       FACT=KG(K,I)/KG(I,I)
       DO 280 J=II,M1
       KG(K,J)=KG(K,J)-FACT*KG(I,J)
280    CONTINUE
```

```
      KG(K,I)=0.0
      BY(K)=BY(K)-FACT*BY(I)
290   CONTINUE
300   CONTINUE
C
C     BACK SUBSTITUE
C
      DO 400 I=1,M1
      II=M1-I+1
      PIVOT=KG(II,II)
      KG(II,II)=0.0
      DO 350 J=II,M1
      BY(II)=BY(II)-KG(II,J)*DY(J)
350   CONTINUE
      DY(II)=BY(II)/PIVOT
400   CONTINUE
C
      RETURN
      END
C=============================================================
C     SUB. OUTPUT THE NODE DISPLACEMENT
C=============================================================
      SUBROUTINE OUTSO1(NE,MCON,NOCP,ROWN,DG,DOF,A,DGX,DGY,X)
C
      INTEGER  NE,DOF,ROWN,NOCP,MCON(30,4),H(4)
      DOUBLE PRECISION DG(ROWN),A(4,4),CP(1,4),DGX(20),DGY(20)
     *                 ,PX(20),PY(20),X(20),U(1,4)
C
      WRITE(6,2007)
2007  FORMAT(/1H,8X,'X',10X,'Y')
C
      DO 120 IE=1,NE
      DO 110 J=1,4
      H(J)=MCON(IE,J)
      PX(J)=X(2*(H(J))-1)
      PY(J)=X(2*(H(J)))
110   CONTINUE
C
      DO 60 J=1,11
      DGX(J)=0.0
      DGY(J)=0.0
      UJ=J/10.0
      UAA=(UJ-0.1)
      U(1,1)=1.0
      U(1,2)=UAA
      U(1,3)=UAA**2.0
      U(1,4)=UAA**3.0
      DO 50 K=1,4
      CP(1,K)=0.0
      DO 50 JJ=1,4
      CP(1,K)=CP(1,K)+U(1,JJ)*A(JJ,K)
50    CONTINUE
C
      DO 55 L=1,4
      DGX(J)=DGX(J)+CP(1,L)*PX(L)
      DGY(J)=DGY(J)+CP(1,L)*PY(L)
55    CONTINUE
```

```
1027          WRITE(6,600)DGX(J),DGY(J)
1028    600   FORMAT(2F12.4)
1029    60    CONTINUE
1030    120   CONTINUE
1031    C
1032          RETURN
1033          END
1034    C=============================================================
```

APPENDIX (C)

Three-dimensional analysis

Appendix C-1

$$r(u,v)=\begin{pmatrix}1 & u & u^2 & u^3\end{pmatrix}DTD^T\begin{pmatrix}1 & v & v^2 & v^3\end{pmatrix}^T$$

The covariant base vectors are defined as,

$a_1=r_{,1}$ $\quad a_2=r_{,2}$ $\quad$ and $a_3=\dfrac{a_1 X a_2}{|a_1 X a_2|}$

henc;

$$a_1=\begin{pmatrix}0 & 1 & 2u & 3u^2\end{pmatrix}DTD^T\begin{pmatrix}1 & v & v^2 & v^3\end{pmatrix}^T$$

$$a_2=\begin{pmatrix}1 & u & u^2 & u^3\end{pmatrix}DTD^T\begin{pmatrix}0 & 1 & 2v & 3v^2\end{pmatrix}^T$$

$$a_{1,1}=\begin{pmatrix}0 & 1 & 2u & 3u^2\end{pmatrix}DTD^T\begin{pmatrix}1 & v & v^2 & v^3\end{pmatrix}^T$$

$$a_{1,2}=\begin{pmatrix}0 & 1 & 2u & 3u^2\end{pmatrix}DTD^T\begin{pmatrix}0 & 1 & 2v & 3v^2\end{pmatrix}^T$$

$$a_{2,2}=\begin{pmatrix}1 & u & u^2 & u^3\end{pmatrix}DTD^T\begin{pmatrix}0 & 1 & 2v & 3v^2\end{pmatrix}^T$$

$$a_{11}=a_1.a_1=V.[D][T]^T[D]^T.U'^T.U'[D][T][D]^T.V^T$$

$$a_{22}=a_2.a_2=V'.[D][T]^T[D]^T.U^T.U[D][T][D]^T.V'^T$$

$$a_{12}=a_{21}=a_1.a_2=V'.[D][T]^T[D]^T.U^T.U'[D][T][D]^T.V^T$$

$$a_3=(a_1 X\ a_2)/\sqrt{a} \qquad a=a_{11}.a_{22}-(a_{12})^2$$

$$b_{11}=a_{1,1}.a_3$$

$$b_{11}=\left[\left(a_1(2).a_2(3)-a_1(3).a_2(2)\right)i-\left(a_1(1).a_2(3)-a_1(3).a_2(1)\right)j\right.$$

$$\left.+\left(a_1(1).a_2(2)-a_1(2).a_2(1)\right)k\right]\frac{a_{1,1}}{\sqrt{a}}$$

$$b_{22}=a_{2,2}.a_3$$

$$b_{22}=\left[\left(a_1(2).a_2(3)-a_1(3).a_2(2)\right)i-\left(a_1(1).a_2(3)-a_1(3).a_2(1)\right)j\right.$$

$$\left.+\left(a_1(1).a_2(2)-a_1(2).a_2(1)\right)k\right]\frac{a_{2,2}}{\sqrt{a}}$$

$$b_{12}=a_{1,2}.a_3$$

$$b_{12}=\left[\left(a_1(2).a_2(3)-a_1(3).a_2(2)\right)i-\left(a_1(1).a_2(3)-a_1(3).a_2(1)\right)j\right.$$

$$\left.+\left(a_1(1).a_2(2)-a_1(2).a_2(1)\right)k\right]\frac{a_{1,2}}{\sqrt{a}}$$

Appendix C-2 (3D computer program list)

3Dprog is ruining in the same way of **2Dprog** except some changes can be written according to the flow diagram shown next. The groups of subroutines indicated as follows;

1. INPDAT to BANDW The structural data is read and stored in appropriate arrays.
2. AANDTA Calculation the matrix [A].
3. FORCE to KFPMATThe element forces are calculated and printed

by cycling throught the elements.

4. VANDTV to FSTR [U] and [V] matrices are calculated and their derevative.

5. SDIFF to TORQT The member stiffness matrix of bending and torsion is assembled.

6. FBGKBG to GSTIF Bending and torsion gloable stiffness matrix.

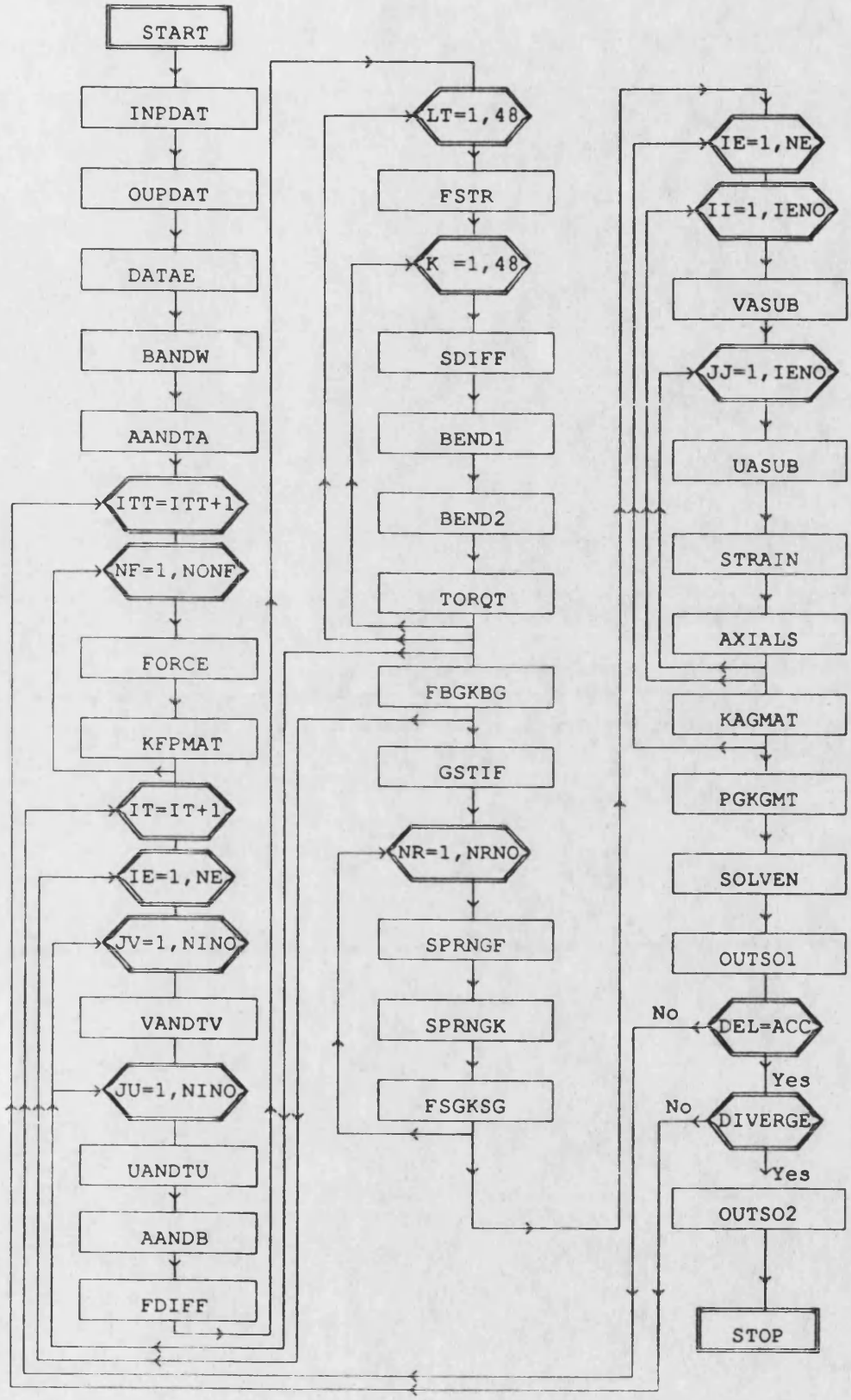
7. SPRNGF to FSGKSG The global stiffness matrix of the spring is calculated by cycling through the boundires.

8. VASUB to KAGMAT The membrane forces of the surfaces is calculated.

9. PGKGMT The global forces stiffness matrix.

10. SOLVE to OUTSO1 The equilibrium equations are solved and the displacement solution is printed.

11. OUTSO2 The coordinates of the final deformed shap are printed.



Flow diagram of 3D-prog.

```
C=============================================================
C                      PROGRAM 3DIMEN.
C
C    THIS PROGRAM IS DEVELOPED TO CARRY OUT THREE-DIMENTIONAL
C    ANALYSIS OF  GRID SHELLS STRUCTURES BY USING  THE FINITE
C   ELEMENT NON-LINEAR TECHNIQUE AND BICUBIC B-SPLINE FUNCTION
C=============================================================
C
C      ------------------------------------------------------
C     |            LIST OF MAIN VARIABLES AND ARRAYS         |
C     |           ....................................       |
C     |                                                      |
C     |  NG      = NUMBER OF ELEMENT GROUPS                  |
C     |  NE      = NUMBER OF ELEMENT                         |
C     |  ROWN    = NUMBER OF ROWS IN THE G-STIFF. MATRIX     |
C     |  COLN    = NUMBER OF COLUMNS IN THE G-STIFF. MATRIX  |
C     |  NOCP    = NUMBER OF CONTROL POINTS                  |
C     |  NCPS    = NUMBER OF NODES PER SPLINE                |
C     |  NDFP    = NUMBER OF DEGREES OF FREEDOM PER POINT    |
C     |  NFNO    = NUMBER OF EXTERNAL FORCES                 |
C     |  NRNO    = NUMBER OF RESTRAIN NODES                  |
C     |  DOF     = NUMBER OF TOTAL DEGREES OF FREEDOM        |
C     |  CORD    = CO-ORDINATES OF THE CONTROL POINTS        |
C     |  MCON    = SPLINE CONNECTIVITY                       |
C     |  PROP    = MATERIAL GROUP & CROSS SECTION PROPERTIES |
C     |  PFR     = EXTERNAL FORCE PROPERTIES                 |
C     |  PFRC    = EXTERNAL FORCE POSITION                   |
C     |  SPR     = SPRNG PROPERTIES                          |
C     |  SPRE    = SPRNG POSITION                            |
C     |  KBE     = ELEMENT BENDING STIFF.                    |
C     |  KAE     = ELEMENT AXIAL STIFF.                      |
C     |  KS      = ELEMENT SPRING STIFF.                     |
C     |  FBE     = ELEMENT BENDING FORCE                     |
C     |  FAE     = ELEMENT AXIAL FORCE                       |
C     |  FS      = ELEMENT SPRING FORCE                      |
C     |  FPE     = ELEMENT APPLIED LOAD                      |
C     |  KBG     = GLOBAL BENDING STIFF.                     |
C     |  KAG     = GLOBAL AXIAL STIFF.                       |
C     |  KSG     = GLOBAL SPRING STIFF.                      |
C     |  FBG     = GLOBAL BENDING FORCE                      |
C     |  FAG     = GLOBAL AXIAL FORCE                        |
C     |  FPG     = GLOBAL EXTERNAL FORCE                     |
C     |  FSG     = GLOBAL SPRING FORCE                       |
C     |  KG      = ELASTIC LINEAR STIFFNESS                  |
C     |  PG      = GLOBAL FORCE MATRIX                       |
C     |  DG      = GLOBAL DISPLACEMENT MATRIX                |
C     |  ALFA    = AXIAL STRAIN                              |
C     |  BETA    = BENDING  STRAIN                           |
C     |  LCNO    = NO. OF CASE OF LOADING                    |
C     |  IT      = ITERATION NUMBER.                         |
C     |  ACCU    = THE ITERATION ACCUARCY                    |
C     |  INCR    = THE LOAD INCREMENTS                       |
C
C=============================================================
```

```
      INTEGER NE,NC,NR,NOCP,ROWN,COLN,LCNO,NF,LN,NINO,IENO,ROWNA
     *     ,NG,IE,NDFP,DOF,NFNO,NRNO,IT,JU,JV,ROWN1,COLN1,COLNA
     *     ,MCON(64,16),SPRE(50,2),PFRC(20,2),NOSE
     *     ,MZZ,NZZ,I,J,N,LT,KT,II,JJ,JS,K,NAN,NCL
C
      REAL CORD(121,3),PROP(20,7),SPR(50,4),DISP(100,3,4,4)
     * ,KBG(300,300),KSG(300,300),KGG(300,300),KG(300,300)
     * ,KS(48,48),KRE(48,8),KQE(8,48),KRG(300,200),KQG(200,300)
     * ,FBG(300),FPG(300),FSG(300),PG(300),INCR(10),PGG(300)
     *,FPE(48),FS(48),PFR(20,4),DG(300),DCC(300),X(121),Y(121)
     * ,KP11(48,48),KP22(48,48),KP33(48,48),PT(300),Z(121)
     * ,QP11(48),QP22(48),QP33(48),FB11(300),FB22(300),FB33(300)
     * ,KB11(300,300),KB22(300,300),KB33(300,300)
     * ,UA,U(1,4),U1(1,4),U2(1,4),VA,V(4,1),V1(4,1),V2(4,1),RX,RY,RZ
     * ,H(1,4),H1(1,4),H2(1,4),VR(4,1),VR1(4,1),VR2(4,1),ACU(300)
     * ,A,AS,AI11,AI22,STR(8),STRG(200),AS11,AS22,PP(16),PCC(300)
     * ,A11,A22,A12,A1(3),A2(3),AA11(3),AA12(3),ADA(4,4)
     * ,FDELT(16),TX211,TS211,TX122,TS122,SPX,SPY,XLEN,YLEN,TENG(200
     * ,ACCU,ALFA,BETA1,BETA2,GAMA,B11,B12,AB11,AT211,AT122,BX22,BS2
     * ,DA1(16,3,3),DA2(16,3,3),DA11(16,3),DAS(16,3),DDA12(16,3,16,3
     * ,DA12(16,3),DAA11(16,3,3),DAA12(16,3,3),DA(16,3)
     * ,DB11(16,3),DB12(16,3),DAB11(16,3),NORU(3),DDNORU(16,3,16,3,3
     * ,DTX11(16,3),DTS11(16,3),DTX22(16,3),DTS22(16,3),DAA22(16,3,3
     * ,DDB22(16,3,16,3),DDA11(16,3,16,3),DDB11(16,3,16,3)
     * ,DDAS(16,3,16,3),DA22(16,3),DNORU(16,3,3),ABV11,ABV22
     * ,BX11,BS11,DBX11(16,3),DBS11(16,3),TPSQ,DLSQ
     * ,B22,AA22(3),AB22,DB22(16,3),DAB22(16,3),DDA22(16,3,16,3)
     * ,DABV11(16,3),DAT122(16,3)
     * ,DABV22(16,3),DAT211(16,3)
     * ,DAS11(16,3),DAS22(16,3)
     * ,DBX22(16,3),DBS22(16,3),TX(200),TY(200)
     * ,UO(1,4),UO1(1,4),VO(4,1),VO1(4,1),AO11,AO22
     * ,HO(1,4),HO1(1,4),VRO(4,1),VRO1(4,1),AO1(3),AO2(3)
C=============================================================
      CHARACTER*10 INFILE
      PRINT *, "type the input file"
      READ(*,100)INFILE
100   FORMAT(A10)
      OPEN(9,FILE=INFILE,FORM='FORMATTED')
C=============================================================
      CALL INPDAT(NG,NE,NR,NC,NOCP,NDFP,DOF,PFR,PFRC,LCNO
     *   ,INCR,PROP,SPRE,SPR,NFNO,NRNO,BETA1,BETA2,GAMA
     *   ,NINO,IENO,SPX,SPY,XLEN,YLEN,AI11,AI22,NOSE,TX,TY,TXM,TYM)
      CALL OUPDAT(NG,NE,NR,NC,NOCP,NDFP,DOF,PROP,SPRE,SPR,NFNO,NRNO
     *   ,PFR,PFRC,LCNO,INCR,NINO,IENO,SPX,SPY,XLEN,YLEN
     *   ,NOSE,TX,TY)
      CALL DATAE(NE,NR,NC,MCON,NOCP,CORD,X,Y,Z,SPX,SPY,XLEN,YLEN,DISP)
      CALL BANDW(NDFP,NOCP,NE,IENO,ROWN,COLN,ROWN1,COLN1,ROWNA,COLNA)
      CALL AANDTA(ADA)
C
      DO 230 NF=1,NFNO
      PP(NF)=PFR(NF,3)
      FDELT(NF)=PFR(NF,4)
230   CONTINUE
```

```
109   C
110         SPSQ=0.0
111         ITT=0
112   900   ITT=ITT+1
113         DO 231 NF=1,NFNO
114         IF(ITT.NE.1) GOTO 422
115         PP(NF)=PP(NF)
116         GO TO 231
117   422   PP(NF)=PP(NF)+FDELT(NF)
118   231   CONTINUE
119   C
120         WRITE(6,901)ITT,PP(1)
121   901   FORMAT(1H,'LOAD CASE NO. ',I3,'     LOAD=',F8.3)
122   C
123         DO 30 NF=1,NFNO
124         CALL FORCE(PP,ITT,NF,PFR,PFRC,FPE,ACU,INCR,ADA)
125         CALL KFPMAT(MCON,NFNO,NF,ROWN1,FPE,FPG,PFRC)
126         ACCU=ABS(ACU(NF))
127         IF(ACU(NF).LE.ACU(1))  ACCU=ACU(NF)
128   30    CONTINUE
129   C
130         DO 848 I=1,ROWN
131         PG(I)=0.0
132         DO 848 J=1,COLN
133         KG(I,J)=0.0
134   848   CONTINUE
135   C
136         IT=0
137   400   IT=IT+1
138         WRITE(6,45)IT
139   45    FORMAT(1H,'ITERATION NO.  ',I5)
140   C
141         DO 5770 I=1,ROWN1
142         FB11(I)=0.0
143         FB22(I)=0.0
144         FB33(I)=0.0
145         DO 5770 J=1,COLN1
146         KB11(I,J)=0.0
147         KB22(I,J)=0.0
148         KB33(I,J)=0.0
149   5770  CONTINUE
150   C
151         DO 847 I=1,ROWN1
152         FBG(I)=0.0
153         DO 847 J=1,COLN1
154         KBG(I,J)=0.0
155   847   CONTINUE
156   C
157         DO 10 IE=1,NE
158         DO 5353 I=1,48
159         QP11(I)=0.0
160         QP22(I)=0.0
161         QP33(I)=0.0
162         DO 5353 J=1,48
```

```
163         KP11(I,J)=0.0
164         KP22(I,J)=0.0
165         KP33(I,J)=0.0
166    5353 CONTINUE
167    C
168         NZZ=0
169         DO 11 JV=1,NINO
170         CALL VANDTV(JV,NINO,NZZ,VA,V,V1,V2)
171         MZZ=0
172         DO 12 JU=1,NINO
173         CALL UANDTU(JU,NINO,MZZ,UA,U,U1,U2)
174         CALL AANDB(IE,U,U1,U2,V,V1,V2,H,H1,H2,VR,VR1,VR2,DISP
175        * ,A,AS,A1,A2,A11,A22,A12,AA22,AA11,AA12,B11,B22,B12,NORU
176        * ,ADA,ABV11,ABV22,BX11,BX22,AT211,AT122,AS11,AS22
177        * ,AB11,AB22,BS11,BS22,TX211,TS211,TX122,TS122)
178         CALL FDIFF(IE,H,H1,H2,VR,VR1,VR2,A,AS,A11,A22,A12,AS11
179        * ,B11,B22,B12,A1,A2,AA11,AA22,AA12,DA,DAS,DA1,NORU,AS22
180        * ,DAT211,DAT122,DAB11,DAB22,DA2,DA11,DA22,DA12,DAA11
181        * ,DB11,DB22,DB12,BX11,BS11,BX22,BS22,DAA12,DABV11,DABV22
182        * ,DBX11,DBS11,DBX22,DBS22,DNORU,DAA22,TX211,TS211,DTX11
183        * ,DTS11,TX122,TS122,DTX22,DTS22,DAS11,DAS22)
184    C
185         DO 1400 LT=1,16
186         DO 1400 N=1,3
187         KT=3*(LT-1)+N
188         CALL FSTR(LT,N,KT,BETA1,BETA2,GAMA,AB11,AB22,AT211,AT122
189        *     ,ABV11,ABV22,DAB22,DAT211,DABV11,DAB11,DAT122,DABV22
190        *     ,QP11,QP22,QP33)
191    C
192         DO 1500 K=1,16
193         DO 1500 L=1,3
194         LN=3*(K-1)+L
195         CALL SDIFF(A,AS,DA,DA1,DA2,DAS,DA11,DA12,DA22,DDAS
196        *     ,DDA11,DDA12,DDA22,NORU,DNORU,DDNORU
197        *     ,A11,A12,A22,AS11,AS22,DAS11,DAS22,LT,N,KT,K,L,LN)
198         CALL BEND1(A11,B11,AB11,BETA1,KP11,DA11,DAA11,DB11,DNORU
199        *     ,A22,B22,AB22,DAB11,DDNORU,DDB11,AA11,LT,N,KT,K,L,LN
200        *     ,DA22,DAA22,DB22,DAB22,DDB22,AA22,DDA11,DDA22)
201         CALL BEND2(AS,A11,A1,A2,AT211,AA11,TX211,TS211,LT,N,KT
202        * ,BETA2,KP22,DA11,DA1,DA2,DAA11,DA12,DAS,DAT211,K,L,LN,A22
203        * ,AA22,DTX11,DTS11,DDA11,DDA12,DDAS,TX122,TS122,DA22,DAA22
204        * ,DTX22,DTS22,DAT122,AT122,A12,DDA22)
205         CALL TORQ(AS,A12,AA12,B12,GAMA,KP33,DA12,DAS,A11,B11
206        * ,A22,B22,DNORU,DAA12,DB12,DDA12,DDAS,DDNORU,LT,N,KT,K,L,LN
207        * ,BS11,BX11,DB11,DDB11,DDA11,DABV11,DBX11,DBS11,DA11,ABV11
208        * ,BS22,BX22,DB22,DDB22,DDA22,DABV22,DBX22,DBS22,DA22,ABV22)
209    1500 CONTINUE
210    1400 CONTINUE
211    C
212         MZZ=MZZ+1
213    12   CONTINUE
214         NZZ=NZZ+1
215    11   CONTINUE
216         CALL FBGKBG(IE,MCON,QP11,QP22,QP33,FB11,FB22,FB33
```

```
      *            ,KP11,KP22,KP33,KB11,KB22,KB33)
10    CONTINUE
      CALL GSTIF(FBG,FB11,FB22,FB33,KBG,KB11,KB22,KB33,ROWN1,COLN1)
C
      DO 502 I=1,ROWN1
      FSG(I)=0.0
      DO 502 J=1,COLN1
      KSG(I,J)=0.0
502   CONTINUE
C
      DO 20 NR=1,NRNO
      CALL SPRNGF(NR,MCON,NRNO,ADA,SPRE,SPR,FS,DISP)
      CALL SPRNGK(NR,MCON,NRNO,ADA,SPRE,SPR,KS)
      CALL FSGKSG(NR,MCON,NRNO,ROWN1,COLN1,FS,FSG,KS,KSG,SPRE)
20    CONTINUE
C
      DO 3050 M=1,ROWNA
      STRG(M)=0.0
      DO 3050 N=1,COLN1
      KQG(M,N)=0.0
      KRG(N,M)=0.0
3050  CONTINUE
C
      DO 55 IE=1,NE
      DO 32 L=1,8
      STR(L)=0.0
      DO 32 KT=1,48
      KRE(KT,L)=0.0
      KQE(L,KT)=0.0
32    CONTINUE
C
      KZ=0
      DO 65 II=1,IENO
      CALL VASUB(II,IENO,KZ,VO,VO1)
      JS=0
      DO 75 JJ=1,IENO
      CALL UASUB(JJ,IENO,JS,UO,UO1)
      CALL STRAIN(IE,ADA,DISP,UO,UO1,VO,VO1,II,JJ,AI11,AI22,AO11
     *            ,AO22,STR,HO,HO1,VRO,VRO1,AO1,AO2,IENO)
      CALL AXIALS(IE,KQE,KRE,II,JJ,AI11,AI22,HO,HO1,VRO,VRO1,AO1
     *      'AO2,IENO,AO11,AO22)
      JS=JS+1
75    CONTINUE
      KZ=KZ+1
65    CONTINUE
      CALL KAGMAT(IE,IENO,MCON,KRE,KRG,KQE,KQG,STR,STRG)
55    CONTINUE
C
      CALL PGKGMT(FBG,FSG,FPG,PG,KBG,KRG,KQG,KSG,KG,ROWN,COLN
     *   ,ROWN1,COLN1,ROWNA,COLNA,KGG,PGG,STRG,TENG,PT)
C
C      CALL SOLVE(ROWN,COLN,KG,PG,DG,PCC)
      CALL SOLVEN(ROWN,COLN,KG,PG,DG,PCC,DCC)
      WRITE(25,911)ITT,PP(1)
```

```
911   FORMAT(1H,'LOAD CASE NO.',I3,'     LOAD=',F8.3)
      CALL OUTSO1(NE,NOCP,IENO,ROWN1,DG,X,Y,Z,TX,TY,TENG
     *           ,MCON,CORD,DISP)
      TPSQ=0.0
      DO 300 I=1,ROWN1
      TPSQ=TPSQ+(PG(I)**2.0)
300   CONTINUE
      DLSQ=0.0
      DO 301 II=1,ROWNA
      I=ROWN1+II
      DLSQ=DLSQ+(PG(I)**2.0)
301   CONTINUE
      PRINT*,"TOTAL OF THE SQU. PG ",TPSQ
      PRINT*,"TOTAL OF THE SQU. STR",DLSQ
C      IF(TPSQ.GE.2.0) GO TO 400
      IF(DLSQ.LE.1.0E-7) GO TO 530
      GO TO 400
530   IF(ITT.LT.LCNO) GO TO 900
C
      CALL OUTSO2(NE,MCON,NOCP,DG,DOF,X,Y,Z,RX,RY,RZ,ADA,CORD)
C
      STOP
      END
C=============================================================
C     SUBROUTINE IN PUT DATA
C=============================================================
      SUBROUTINE INPDAT(NG,NE,NR,NC,NOCP,NDFP,DOF,PFR,PFRC,LCNO
     *   ,INCR,PROP,SPRE,SPR,NFNO,NRNO,BETA1,BETA2,GAMA
     *   ,NINO,IENO,SPX,SPY,XLEN,YLEN,AI11,AI22,NOSE,TX,TY,TXM,TYM)
C
      INTEGER  NG,NE,NOCP,NR,NC,NDFP,DOF,NFNO,NRNO,LCNO,IENO
     *        ,I,J,II,M,N,NF,NINO,SPRE(50,2),PFRC(20,2),NOSE
      REAL  PROP(20,7),SPR(50,4),PFR(20,4),INCR(10),AI11,AI22,TXM
     * ,TYM,BETA1,BETA2,GAMA,SPX,SPY,XLEN,YLEN,TX(200),TY(200)
C
      READ(9,*)NG,NE,NDFP,NFNO,NRNO,NR,NC,SPX,SPY
      NOCP=NR*NC
      DOF =NOCP*NDFP
      XLEN=(NC-3)*SPX
      YLEN=(NR-3)*SPY
C
      DO 800 II=1,NG
      READ(9,*) M,(PROP(II,J),J=1,7)
      AI11=PROP(II,2)*PROP(II,2)
      AI22=PROP(II,2)*PROP(II,2)
      BETA1=PROP(II,3)*PROP(II,5)/((PROP(II,2))**3.0)
      BETA2=PROP(II,3)*PROP(II,6)/((PROP(II,2))**3.0)
      GAMA =PROP(II,4)*PROP(II,7)
800   CONTINUE
      READ(9,*)NINO,IENO,LCNO
      NOSE=NE*(IENO**2)
      READ(9,*) (M,(PFRC(M,N),N=1,2),(PFR(M,N),N=1,4),M=1,NFNO)
      READ(9,*) (M,(SPRE(M,N),N=1,2),(SPR(M,N),N=1,4),M=1,NRNO)
      READ(9,*) (TXM,TYM)
```

```
      DO 12 M=1,NOSE
      TX(M)=TXM
      TY(M)=TYM
12    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUBROUTINE OUT PUT DATA
C=============================================================
      SUBROUTINE OUPDAT(NG,NE,NR,NC,NOCP,NDFP,DOF,PROP,SPRE,SPR
     *     ,NFNO,NRNO,PFR,PFRC,LCNO,INCR,NINO,IENO,SPX,SPY,XLEN
     *     ,YLEN,NOSE,TX,TY)
C
      INTEGER    NG,NE,NOCP,NDFP,DOF,NFNO,NRNO,LCNO,IENO
     *          ,NR,NC,NINO,SPRE(50,2),PFRC(20,2),NOSE
      REAL       PROP(20,7),SPR(50,4),INCR(10),PFR(20,4)
     *          ,SPX,SPY,XLEN,YLEN,TX(200),TY(200)
C
      WRITE(25,700)
700   FORMAT(//22X,'THE DATA'/22X,'========')
      WRITE(25,710)NG,NE,NOCP,NR,NC,NDFP,DOF,NFNO,NRNO
710   FORMAT(1H,'DATA SUPPLIED',//1H,'NUMBER OF ELE.GROUP=',I5,
     *           /1H,'TOTAL NUMBER OF ELEMEMT =',I5,
     *           /1H,'NUMBER OF CONTROL POINT =',I5,
     *           /1H,'NO. OF NODES PER ROW    =',I5,
     *           /1H,'NO. OF NODES PER COL.   =',I5
     *           /1H,'NUM. OF D.O.F PER N.C.P =',I5,
     *           /1H,'TOTAL NUM. OF D.O.F     =',I5,
     *           /1H,'NUMBER OF NODEL FORCES  =',I5,
     *           /1H,'NUMBER OF SPRING        =',I5)
      WRITE(25,720)XLEN,YLEN
720   FORMAT(//7X,'X-LENGTH=',F5.2,'      Y-LENGTH=',F5.2)
      WRITE(25,760)
760   FORMAT(//1H,'MEMBER PROP.',/1H,'MEM.',5X,'(AREA)',5X,'(LEN)'
     *,5X,'(E)',5X,'(G)',5X,'(Ix)',5X,'(Iy)',9X,'(J)')
      WRITE(25,770)(M,(PROP(M,N),N=1,7),M=1,NG)
770   FORMAT(1H,I2,E12.3,6E10.3)
      WRITE(25,766)NINO,IENO,LCNO
766   FORMAT(/1H,'NO. OF NUMER. INTEGRATION =',I5,
     *       /1H,'NO. OF INTEGRATED EL/ELEM.=',I5,
     *       /1H,'NO. OF LOADING CASE       =',I5)
      WRITE(25,776)
776   FORMAT(//1H,'APPLIED LOADS',/1H,'LOAD NO.',4X,
     *'ELEM.NO.',4X,'DIREC.',10X,'(U)',6X,'(V)',8X,'(PP)',5X,'INCR')
      WRITE(25,777)(M,(PFRC(M,N),N=1,2),(PFR(M,N),N=1,4),M=1,NFNO)
777   FORMAT(1H,I5,2I10,F19.3,3F10.3)
      WRITE(25,780)
780   FORMAT(//1H,'SPRING PROPERTIES',/1H,'SPR. NO.',3X,
     *'ELEM.NO.',3X,'DIREC.',5X,'(U)',6X,'(V)',10X,'(S)',8X,'CONST.')
      WRITE(25,790)(M,(SPRE(M,N),N=1,2),(SPR(M,N),N=1,4),M=1,NRNO)
790   FORMAT(1H,I5,2I10,F11.3,F10.3,E14.3,F12.3)
      WRITE(25,795)(M,TX(M),TY(M),M=1,NOSE)
795   FORMAT(I10,2E13.4)
```

```
C
      RETURN
      END
C=================================================================
C     SUBROUTINE  DATA FOR CALCU. THE COORD. AND ELEM. CONEC.
C=================================================================
      SUBROUTINE DATAE(NE,NR,NC,MCON,NOCP,CORD,X,Y,Z,SPX,SPY,XLEN
     *          ,YLEN,DISP)
C
      INTEGER   NE,NOCP,NR,NC,MCON(64,16),NJ(4),MF(100,4,4)
      REAL      X(121),Y(121),Z(121),ZZ(121),CORD(121,3),ALF,BET
     *         ,SPX,SPY,XLEN,YLEN,DISP(100,3,4,4),XT(121)
C
      ALF=30
      BET=45
      WRITE(25,*)ALF,BET,NE,NOCP
C      WRITE(25,740)
C740   FORMAT(//1H,'MEMBER CONNECT.',//,'ELEM NO.',20X,'N1.N16')
      J=0
      DO 10 K=1,NE
      JX=0
      DO 100 L=1,4
      LL=L-1
      JY=0
      DO 200 M=1,4
      NJ(M)=K+J+JY+(NC*LL)
      LK=(4*JX)+JY+1
      MCON(K,LK)=NJ(M)
      JY=JY+1
200   CONTINUE
      JX=JX+1
100   CONTINUE
      WRITE(25,38)K,(MCON(K,I),I=1,16)
38    FORMAT(17I4)
      NL=NC-3
      ML=NR-3
      DO 11 MX=1,ML
      MN=MX*NL
      IF(K.EQ.MN) GO TO 16
11    CONTINUE
      GO TO 10
16    J=J+3
10    CONTINUE
C
C      WRITE(25,720)
C720   FORMAT(//1H,'NODES COORDINATE',/1H,8X,'NODE',9X,
C     *          '(X)',10X,'(Y)',10X,'(Z)')
      XMX=-(XLEN/2+2*SPX)
      YMY=-(YLEN/2+SPY)
      DO 121 KK=1,NOCP
      X(KK)=XMX+SPX
      XT(KK)=ABS(X(KK))
      IF(XT(KK).LT.1.0E-04) X(KK)=0.0
      Y(KK)=YMY
```

```
433         AS=0.5
434         BS=0.5
435         ZZ(KK)=-AS*(COSH(X(KK)/AS))-BS*(COSH(Y(KK)/BS))
436         Z(KK)=ZZ(KK)-ZZ(1)+0.1
437         CORD(KK,1)=X(KK)
438         CORD(KK,2)=Y(KK)
439         CORD(KK,3)=Z(KK)
440         IF(KK.GE.NC) THEN
441         DO 62 L=1,NC
442         NA=NC*L
443         IF(KK.EQ.NA) GO TO 19
444         GO TO 62
445   19       XMX=-XLEN/2-2*SPX
446            YMY=YMY+SPY
447            GO TO 121
448   62       CONTINUE
449         ENDIF
450         XMX=XMX+SPX
451   121   CONTINUE
452   C
453         WRITE(25,3090)(I,(CORD(I,J),J=1,3),I=1,NOCP)
454   3090  FORMAT(I5,3E13.4)
455         DO 401 IE=1,NE
456         DO 400 I=1,4
457         DO 400 J=1,4
458         LL=4*(I-1)+J
459         LT=MCON(IE,LL)
460         DISP(IE,1,I,J)=CORD(LT,1)
461         DISP(IE,2,I,J)=CORD(LT,2)
462         DISP(IE,3,I,J)=CORD(LT,3)
463   400   CONTINUE
464   401   CONTINUE
465   C
466         RETURN
467         END
468   C=============================================================
469   C     SUB. TO CALCULAT THE BAND WIDTH OF THE STIFF. MATRICX
470   C=============================================================
471         SUBROUTINE BANDW(NDFP,NOCP,NE,IENO,ROWN,COLN,ROWN1,COLN1
472        *                ,ROWNA,COLNA)
473   C
474         INTEGER NE,IENO,NDFP,COLN,ROWN,NOCP,ROWN1,COLN1,ROWNA,COLNA
475   C
476         ROWN1=NDFP*NOCP
477         COLN1=NDFP*NOCP
478         ROWNA=2*(IENO**2)*NE
479         COLNA=2*(IENO**2)*NE
480         ROWN=ROWN1+COLNA
481         COLN=COLN1+COLNA
482   C
483         RETURN
484         END
485   C=============================================================
486   C     SUB. FOR CALCULATE THE A MATRIX
```

```
C=============================================================
      SUBROUTINE AANDTA(ADA)
C
      REAL  ADA(4,4),DD
C
      DD=1.0/6
      ADA(1,1)= 1*DD
      ADA(1,2)= 4*DD
      ADA(1,3)= 1*DD
      ADA(2,1)=-3*DD
      ADA(2,3)= 3*DD
      ADA(3,1)= 3*DD
      ADA(3,2)=-6*DD
      ADA(3,3)= 3*DD
      ADA(4,1)=-1*DD
      ADA(4,2)= 3*DD
      ADA(4,3)=-3*DD
      ADA(4,4)= 1*DD
C
      RETURN
      END
C=============================================================
C     SUB. FOR CALCULATING THE EQUIVLANT FORCE ON THE C.P.
C=============================================================
      SUBROUTINE FORCE(PP,ITT,NF,PFR,PFRC,FPE,ACU,INCR,ADA)
C
      INTEGER   NF,ITT,PFRC(20,2)
      REAL INCR(10),PFR(20,4),FPE(48),ADA(4,4),DELP(10)
     *,ACU(20),U(1,4),UU(10),PP(16),C(1,4),H(4,1),V(4,1),VV(10)
C
      DO 10 I=1,48
      FPE(I)=0.0
10    CONTINUE
C
C      DELP(NF)=INCR(NF)
C      IF(ITT.EQ.1) DELP(NF)=0.0
C      PP(NF) =PP(NF)+DELP(NF)
      UU(NF) =PFR(NF,1)
      VV(NF) =PFR(NF,2)
      ACU(NF)=PP(NF)/100.0
      IF(PP(NF).EQ.0.0) ACU(NF)=0.1
      U(1,1)=1.0
      U(1,2)=UU(NF)
      U(1,3)=UU(NF)**2.0
      U(1,4)=UU(NF)**3.0
      V(1,1)=1.0
      V(2,1)=VV(NF)
      V(3,1)=VV(NF)**2.0
      V(4,1)=VV(NF)**3.0
      DO 300 L=1,4
      C(1,L)=0.0
      H(L,1)=0.0
      DO 30 K=1,4
      C(1,L)=C(1,L)+U(1,K)*ADA(K,L)
```

```
541         H(L,1)=H(L,1)+ADA(K,L)*V(K,1)
542   30    CONTINUE
543   300   CONTINUE
544   C
545         DO 40 I=1,4
546         DO 40 J=1,4
547         LL=4*(I-1)+J
548         IF(PFRC(NF,2).EQ.3) JS=3*LL
549         IF(PFRC(NF,2).EQ.2) JS=3*LL-1
550         IF(PFRC(NF,2).EQ.1) JS=3*LL-2
551         FPE(JS)=FPE(JS)+PP(NF)*C(1,J)*H(I,1)
552   40    CONTINUE
553   C
554         RETURN
555         END
556   C=============================================================
557   C     SUB. FOR CALCULATING THE GLOBAL FORCE  MATRCIS
558   C=============================================================
559         SUBROUTINE KFPMAT(MCON,NFNO,NF,ROWN1,FPE,FPG,PFRC)
560   C
561         INTEGER  NFNO,NF,ROWN1,MCON(64,16),PFRC(20,2),IR(16)
562         REAL   FPE(48),FPG(300)
563   C
564         IF(NF.NE.1) GO TO 75
565         DO 70 I=1,ROWN1
566         FPG(I)=0.0
567   70    CONTINUE
568   C
569   75    MF=PFRC(NF,1)
570         DO 35 J=1,16
571         IR(J)=MCON(MF,J)
572   35    CONTINUE
573   C
574         DO 45 I=1,16
575         DO 45 J=1,3
576         IQ=3*(IR(I)-1)+J
577         JQ=3*(I-1)+J
578         FPG(IQ)=FPG(IQ)+FPE(JQ)
579   45    CONTINUE
580   C
581         RETURN
582         END
583   C=============================================================
584   C     SUBROUTINE VANDA TO CALCU. [V] MATRICES.
585   C=============================================================
586         SUBROUTINE VANDTV(JV,NINO,NZZ,VA,V,V1,V2)
587   C
588         INTEGER  JV,NZZ,NINO
589         REAL VA,V(4,1),V1(4,1),V2(4,1)
590   C
591         VA=(JV+NZZ)/(2.0*NINO)
592         V(1,1)=1.0
593         V(2,1)=VA
594         V(3,1)=VA**2.0
```

```
595         V(4,1)=VA**3.0
596   C
597         V1(2,1)=1.0
598         V1(3,1)=2.0*VA
599         V1(4,1)=3.0*(VA**2.0)
600   C
601         V2(3,1)=2.0
602         V2(4,1)=6.0*VA
603   C
604         RETURN
605         END
606   C=============================================================
607   C     SUBROUTINE UANDA TO CALCU. [U] MATRICES.
608   C=============================================================
609         SUBROUTINE UANDTU(JU,NINO,MZZ,UA,U,U1,U2)
610   C
611         INTEGER  JU,MZZ,NINO
612         REAL UA,U(1,4),U1(1,4),U2(1,4)
613   C
614         UA=(JU+MZZ)/(2.0*NINO)
615         U(1,1)=1.0
616         U(1,2)=UA
617         U(1,3)=UA**2.0
618         U(1,4)=UA**3.0
619   C
620         U1(1,2)=1.0
621         U1(1,3)=2.0*UA
622         U1(1,4)=3.0*(UA**2.0)
623   C
624         U2(1,3)=2.0
625         U2(1,4)=6.0*UA
626   C
627         RETURN
628         END
629   C=============================================================
630   C     SUBROUTINE AANDB TO CALCU. [B] MATRIX
631   C=============================================================
632         SUBROUTINE AANDB(IE,U,U1,U2,V,V1,V2,H,H1,H2,VR,VR1,VR2,DISP
633        *  ,A,AS,A1,A2,A11,A22,A12,AA22,AA11,AA12,B11,B22,B12,NORU
634        *  ,ADA,ABV11,ABV22,BX11,BX22,AT211,AT122,AS11,AS22
635        *  ,AB11,AB22,BS11,BS22,TX211,TS211,TX122,TS122)
636   C
637         INTEGER IE
638         REAL ADA(4,4),C(1,4),C1(1,4),C2(1,4)
639        *   ,DISP(100,3,4,4),U(1,4),U1(1,4),U2(1,4),VR(4,1),VR1(4,1)
640        *   ,V(4,1),V1(4,1),V2(4,1),H(1,4),H1(1,4),H2(1,4),VR2(4,1)
641        *   ,A,AS,A1(3),A2(3),AS11,AS22,A11,A22,A12,AA11(3),AA12(3)
642        *   ,B11,B12,AB11,AT211,ABV11,ABV22,BX11,BX22,NORU(3),AA22(3)
643        *   ,B22,AB22,TX211,TS211,TX122,TS122,BS11,BS22,AT122
644   C
645         DO 20 L=1,4
646         H(1,L) =0.0
647         H1(1,L)=0.0
648         H2(1,L)=0.0
```

```
649           VR(L,1) =0.0
650           VR1(L,1)=0.0
651           VR2(L,1)=0.0
652           DO 203 K=1,4
653           H(1,L) =H(1,L)+U(1,K)*ADA(K,L)
654           H1(1,L)=H1(1,L)+U1(1,K)*ADA(K,L)
655           H2(1,L)=H2(1,L)+U2(1,K)*ADA(K,L)
656           VR(L,1) =VR(L,1)+ADA(K,L)*V(K,1)
657           VR1(L,1)=VR1(L,1)+ADA(K,L)*V1(K,1)
658           VR2(L,1)=VR2(L,1)+ADA(K,L)*V2(K,1)
659     203   CONTINUE
660     20    CONTINUE
661     C
662           A11=0.0
663           A22=0.0
664           A12=0.0
665           DO 200 N=1,3
666           A1(N)=0.0
667           A2(N)=0.0
668           AA11(N)=0.0
669           AA22(N)=0.0
670           AA12(N)=0.0
671           DO 202 I=1,4
672           C(1,I)=0.0
673           C1(1,I)=0.0
674           C2(1,I)=0.0
675           DO 202 J=1,4
676           C(1,I) =C(1,I) +H(1,J)*DISP(IE,N,J,I)
677           C1(1,I)=C1(1,I)+H1(1,J)*DISP(IE,N,J,I)
678           C2(1,I)=C2(1,I)+H2(1,J)*DISP(IE,N,J,I)
679     202   CONTINUE
680     C
681           DO 201 I=1,4
682           A1(N)=A1(N)+C(1,I)*VR1(I,1)
683           A2(N)=A2(N)+C1(1,I)*VR(I,1)
684           AA11(N)=AA11(N)+C(1,I)*VR2(I,1)
685           AA22(N)=AA22(N)+C2(1,I)*VR(I,1)
686           AA12(N)=AA12(N)+C1(1,I)*VR1(I,1)
687     201   CONTINUE
688           A11=A11+A1(N)*A1(N)
689           A22=A22+A2(N)*A2(N)
690           A12=A12+A1(N)*A2(N)
691     200   CONTINUE
692     C
693           AS11=A11**0.5
694           AS22=A22**0.5
695           A =A11*A22-(A12**2.0)
696           AS=A**(0.5)
697           NORU(1)=A1(2)*A2(3)-A1(3)*A2(2)
698           NORU(2)=A1(3)*A2(1)-A1(1)*A2(3)
699           NORU(3)=A1(1)*A2(2)-A1(2)*A2(1)
700           B11=0.0
701           B22=0.0
702           B12=0.0
```

```
      TX211=0.0
      TX122=0.0
C
      DO 100 N=1,3
      NORU(N)=NORU(N)/AS
      B11=B11+AA11(N)*NORU(N)
      B22=B22+AA22(N)*NORU(N)
      B12=B12+AA12(N)*NORU(N)
      TX211=TX211+(AA11(N)*(A11*A2(N)-A12*A1(N)))
      TX122=TX122+(AA22(N)*(A12*A2(N)-A22*A1(N)))
100   CONTINUE
      TS211=1.0/(AS*A11)
      TS122=1.0/(AS*A22)
      BX11=A12*B11-A11*B12
      BX22=A22*B12-A12*B22
      BS11=1.0/(AS*A11)
      BS22=1.0/(AS*A22)
C
      AB11 =B11/A11
      AB22 =B22/A22
      AT211=TX211*TS211
      AT122=TX122*TS122
      ABV11=-BX11*BS11
      ABV22=-BX22*BS22
C
      RETURN
      END
C=============================================================
C     SUBROUTINE TO DIFF. a,a1,a2,a11,a22,.. regeard to displ.
C=============================================================
      SUBROUTINE FDIFF(IE,H,H1,H2,VR,VR1,VR2,A,AS,A11,A22,A12,AS11
     *  ,B11,B22,B12,A1,A2,AA11,AA22,AA12,DA,DAS,DA1,NORU,AS22
     *  ,DAT211,DAT122,DAB11,DAB22,DA2,DA11,DA22,DA12,DAA11
     *  ,DB11,DB22,DB12,BX11,BS11,BX22,BS22,DAA12,DABV11,DABV22
     *  ,DBX11,DBS11,DBX22,DBS22,DNORU,DAA22,TX211,TS211,DTX11
     *  ,DTS11,TX122,TS122,DTX22,DTS22,DAS11,DAS22)
C
      INTEGER IE,I,J,LX,LY,LT
      REAL H(1,4),H1(1,4),H2(1,4),BX11,BS11,AS11,AS22
     *,VR(4,1),VR1(4,1),VR2(4,1),A,A11,A22,A12,AS,B11,B12,BX22,BS22
     *,A1(3),A2(3),AA11(3),AA12(3),DAA11(16,3,3),DAA22(16,3,3)
     *,DA1(16,3,3),DA2(16,3,3),DA11(16,3),DA(16,3),DAA12(16,3,3)
     *,DAS(16,3),DA22(16,3),DA12(16,3),DAB11(16,3)
     *,DB11(16,3),DNORU(16,3,3),DB12(16,3),NORU(3)
     *,DTX11(16,3),DTS11(16,3),DAT211(16,3),DBS11(16,3)
     *,DBX11(16,3),DB22(16,3),DAB22(16,3),AA22(3),B22
     *,TX122,TS122,DTX22(16,3),DTS22(16,3),DAT122(16,3)
     *,TX211,TS211,DBX22(16,3),DBS22(16,3),DABV11(16,3),DABV22(16,3)
     *,DAS11(16,3),DAS22(16,3)
C
      DO 2200 I=1,4
      DO 2200 J=1,4
      LT=4*(I-1)+J
      DO 2400 N=1,3
```

```
C
      DA11(LT,N)=0.0
      DA22(LT,N)=0.0
      DA12(LT,N)=0.0
      DO 30 M=1,3
      IF(M.NE.N) GO TO 35
      DA1(LT,N,M)=H(1,I)*VR1(J,1)
      DA2(LT,N,M)=H1(1,I)*VR(J,1)
      DAA11(LT,N,M)=H(1,I)*VR2(J,1)
      DAA22(LT,N,M)=H2(1,I)*VR(J,1)
      DAA12(LT,N,M)=H1(1,I)*VR1(J,1)
      GO TO 30
35    DA1(LT,N,M)=0.0
      DA2(LT,N,M)=0.0
      DAA11(LT,N,M)=0.0
      DAA22(LT,N,M)=0.0
      DAA12(LT,N,M)=0.0
30    CONTINUE
C
      DNORU(LT,N,1)=DA1(LT,N,2)*A2(3)+A1(2)*DA2(LT,N,3)
     *              -DA1(LT,N,3)*A2(2)-A1(3)*DA2(LT,N,2)
      DNORU(LT,N,2)=DA1(LT,N,3)*A2(1)+A1(3)*DA2(LT,N,1)
     *              -DA1(LT,N,1)*A2(3)-A1(1)*DA2(LT,N,3)
      DNORU(LT,N,3)=DA1(LT,N,1)*A2(2)+A1(1)*DA2(LT,N,2)
     *              -DA1(LT,N,2)*A2(1)-A1(2)*DA2(LT,N,1)
C
      DO 50 M=1,3
      DA11(LT,N)=DA11(LT,N)+(2*A1(M)*DA1(LT,N,M))
      DA22(LT,N)=DA22(LT,N)+(2*A2(M)*DA2(LT,N,M))
      DA12(LT,N)=DA12(LT,N)+(A1(M)*DA2(LT,N,M)+A2(M)*DA1(LT,N,M))
50    CONTINUE
      DAS11(LT,N)=0.5*DA11(LT,N)/AS11
      DAS22(LT,N)=0.5*DA22(LT,N)/AS22
      DA(LT,N)=A11*DA22(LT,N)+A22*DA11(LT,N)-2.0*A12*DA12(LT,N)
      DAS(LT,N)=DA(LT,N)/(2.0*AS)
C
      DB11(LT,N)=0.0
      DB22(LT,N)=0.0
      DB12(LT,N)=0.0
      DTX11(LT,N)=0.0
      DTX22(LT,N)=0.0
      DO 60 M=1,3
      DNORU(LT,N,M)=(DNORU(LT,N,M)*AS-DAS(LT,N)*NORU(M))/A
      DB11(LT,N)=DB11(LT,N)+(NORU(M)*DAA11(LT,N,M)
     *                      + AA11(M)*DNORU(LT,N,M))
      DB22(LT,N)=DB22(LT,N)+(NORU(M)*DAA22(LT,N,M)
     *                      + AA22(M)*DNORU(LT,N,M))
      DB12(LT,N)=DB12(LT,N)+(NORU(M)*DAA12(LT,N,M)
     *                      + AA12(M)*DNORU(LT,N,M))
C
      DTX11(LT,N)=DTX11(LT,N)+(A11*DA2(LT,N,M)+DA11(LT,N)*A2(M)
     *               -A12*DA1(LT,N,M)-A1(M)*DA12(LT,N))*AA11(M)
     *               +DAA11(LT,N,M)*(A11*A2(M)-A12*A1(M))
      DTX22(LT,N)=DTX22(LT,N)+(A12*DA2(LT,N,M)+A2(M)*DA12(LT,N)
```

```
811         *                -A22*DA1(LT,N,M)-A1(M)*DA22(LT,N))*AA22(M)
812         *                +DAA22(LT,N,M)*(A12*A2(M)-A22*A1(M))
813   60     CONTINUE
814   C
815         DTS11(LT,N)=-(A11*DAS(LT,N)+AS*DA11(LT,N))/((A11*AS)**2.0)
816         DTS22(LT,N)=-(A22*DAS(LT,N)+AS*DA22(LT,N))/((A22*AS)**2.0)
817         DBX11(LT,N)=DB11(LT,N)*A12+B11*DA12(LT,N)
818         *             -DB12(LT,N)*A11-B12*DA11(LT,N)
819         DBX22(LT,N)=DB12(LT,N)*A22+B12*DA22(LT,N)
820         *             -DB22(LT,N)*A12-B22*DA12(LT,N)
821         DBS11(LT,N)=-(DAS(LT,N)*A11+AS*DA11(LT,N))/((A11*AS)**2.0)
822         DBS22(LT,N)=-(DAS(LT,N)*A22+AS*DA22(LT,N))/((A22*AS)**2.0)
823   C
824         DAB11(LT,N)=(A11*DB11(LT,N)-B11*DA11(LT,N))/(A11**2.0)
825         DAB22(LT,N)=(A22*DB22(LT,N)-B22*DA22(LT,N))/(A22**2.0)
826         DAT211(LT,N)=(TS211*DTX11(LT,N)+TX211*DTS11(LT,N))
827         DAT122(LT,N)=(TS122*DTX22(LT,N)+TX122*DTS22(LT,N))
828         DABV11(LT,N)=-(BS11*DBX11(LT,N)+BX11*DBS11(LT,N))
829         DABV22(LT,N)=-(BS22*DBX22(LT,N)+BX22*DBS22(LT,N))
830   2400   CONTINUE
831   2200   CONTINUE
832   C
833         RETURN
834         END
835   C=============================================================
836   C      SUBROUTINE TO CALCULAT FORCES FROM V.WORK
837   C=============================================================
838         SUBROUTINE FSTR(LT,N,KT,BETA1,BETA2,GAMA,AB11,AB22,AT211,AT122
839         *           ,ABV11,ABV22,DAB22,DAT211,DABV11,DAB11,DAT122,DABV22
840         *           ,QP11,QP22,QP33)
841   C
842         INTEGER  LT,N,M,KT
843         REAL BETA1,BETA2,GAMA,AB11,AB22,AT211,AT122,ABV11,ABV22
844         *     ,FP11(48),FP22(48),FP33(48),QP11(48),QP22(48),QP33(48)
845         *     ,DAB22(16,3),DAT211(16,3),DABV11(16,3)
846         *     ,DAB11(16,3),DAT122(16,3),DABV22(16,3)
847   C
848         FP11(KT)=BETA1*(AB11*DAB11(LT,N)+AB22*DAB22(LT,N))
849         FP22(KT)=BETA2*(AT211*DAT211(LT,N)+AT122*DAT122(LT,N))
850         FP33(KT)=GAMA*(ABV11*DABV11(LT,N)+ABV22*DABV22(LT,N))
851   C
852         QP11(KT)=QP11(KT)+FP11(KT)
853         QP22(KT)=QP22(KT)+FP22(KT)
854         QP33(KT)=QP33(KT)+FP33(KT)
855   C
856         RETURN
857         END
858   C=============================================================
859   C      SUBROUTINE TO DIFF. a,a1,a2,a11,a22,.. regeard to displ.
860   C=============================================================
861         SUBROUTINE SDIFF(A,AS,DA,DA1,DA2,DAS,DA11,DA12,DA22,DDAS
862         *     ,DDA11,DDA12,DDA22,NORU,DNORU,DDNORU
863         *     ,A11,A12,A22,AS11,AS22,DAS11,DAS22,LT,N,KT,K,L,LN)
864   C
```

```
      INTEGER LT,N,KT,K,L,LN
      REAL A,AS,A11,A12,A22,DA1(16,3,3),DA2(16,3,3),DA11(16,3),DA(16,3
     *    ,DAS(16,3),DA22(16,3),DA12(16,3),DDA12(16,3,16,3),AS11,AS22
     *    ,NORU(3),DNORU(16,3,3),DDNORU(16,3,16,3,3),DDAS(16,3,16,3)
     *    ,DDA11(16,3,16,3),DDA(16,3,16,3)
     *    ,DDA22(16,3,16,3),DAS11(16,3),DAS22(16,3)
C
      DDNORU(LT,N,K,L,1)=DA1(LT,N,2)*DA2(K,L,3)+DA1(K,L,2)*DA2(LT,N,3)
     *                  -DA1(LT,N,3)*DA2(K,L,2)-DA1(K,L,3)*DA2(LT,N,2)
      DDNORU(LT,N,K,L,2)=DA1(LT,N,3)*DA2(K,L,1)+DA1(K,L,3)*DA2(LT,N,1)
     *                  -DA1(LT,N,1)*DA2(K,L,3)-DA1(K,L,1)*DA2(LT,N,3)
      DDNORU(LT,N,K,L,3)=DA1(LT,N,1)*DA2(K,L,2)+DA1(K,L,1)*DA2(LT,N,2)
     *                  -DA1(LT,N,2)*DA2(K,L,1)-DA1(K,L,2)*DA2(LT,N,1)
C
      DDA11(LT,N,K,L)=0.0
      DDA22(LT,N,K,L)=0.0
      DDA12(LT,N,K,L)=0.0
      DO 30 M=1,3
      DDA11(LT,N,K,L)=DDA11(LT,N,K,L)+(2*DA1(LT,N,M)*DA1(K,L,M))
      DDA22(LT,N,K,L)=DDA22(LT,N,K,L)+(2*DA2(LT,N,M)*DA2(K,L,M))
      DDA12(LT,N,K,L)=DDA12(LT,N,K,L)+(DA2(LT,N,M)*DA1(K,L,M)
     *                               +DA1(LT,N,M)*DA2(K,L,M))
30    CONTINUE
      DDA(LT,N,K,L)=DA22(LT,N)*DA11(K,L)+A11*DDA22(LT,N,K,L)
     *             +DA11(LT,N)*DA22(K,L)+A22*DDA11(LT,N,K,L)
     *             -2.0*(DA12(LT,N)*DA12(K,L)+A12*DDA12(LT,N,K,L))
      DDAS(LT,N,K,L)=(DDA(LT,N,K,L)*AS-DA(LT,N)*DAS(K,L))/(2.0*A)
      DO 40 M=1,3
      DDNORU(LT,N,K,L,M)=((DDNORU(LT,N,K,L,M)*AS+DNORU(LT,N,M)*DAS(K,L
     *           -DDAS(LT,N,K,L)*NORU(M)-DAS(LT,N)*DNORU(K,L,M))*A
     *           -(DNORU(LT,N,M)*AS-DAS(LT,N)*NORU(M))*DA(K,L))/(A**2.
40    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUBROUTINE TO DIFF. a,a1,a2,a11,a22,.. regeard to displ.
C=============================================================
      SUBROUTINE BEND1(A11,B11,AB11,BETA1,KP11,DA11,DAA11,DB11,DNORU
     *     ,A22,B22,AB22,DAB11,DDNORU,DDB11,AA11,LT,N,KT,K,L,LN
     *     ,DA22,DAA22,DB22,DAB22,DDB22,AA22,DDA11,DDA22)
C
      INTEGER LT,N,KT,K,L,LN
      REAL A11,A22,BETA1,KP1(48,48),DNORU(16,3,3),DDNORU(16,3,16,3,3)
     *    ,AB11,DB11(16,3),DA11(16,3),DAB11(16,3),DAA11(16,3,3),AA11(3
     *    ,AB22,DB22(16,3),DA22(16,3),DAB22(16,3),DAA22(16,3,3),AA22(3
     *    ,B11,DDB11(16,3,16,3),DDAB11(16,3,16,3)
     *    ,B22,DDB22(16,3,16,3),DDAB22(16,3,16,3)
     *    ,DDA11(16,3,16,3),DDA22(16,3,16,3),KP11(48,48)
C
      DDB11(LT,N,K,L)=0.0
      DDB22(LT,N,K,L)=0.0
      DO 30 M=1,3
      DDB11(LT,N,K,L)=DDB11(LT,N,K,L)+(AA11(M)*DDNORU(LT,N,K,L,M)
```

```
919         *     +DAA11(K,L,M)*DNORU(LT,N,M)+DAA11(LT,N,M)*DNORU(K,L,M))
920          DDB22(LT,N,K,L)=DDB22(LT,N,K,L)+(AA22(M)*DDNORU(LT,N,K,L,M)
921         *     +DAA22(K,L,M)*DNORU(LT,N,M)+DAA22(LT,N,M)*DNORU(K,L,M))
922    30    CONTINUE
923    C
924          DDAB11(LT,N,K,L)=((A11*DDB11(LT,N,K,L)+DA11(K,L)*DB11(LT,N)
925         *         -B11*DDA11(LT,N,K,L)-DB11(K,L)*DA11(LT,N))*(A11**2)
926         *         -(A11*DB11(LT,N)-B11*DA11(LT,N))*2*A11*DA11(K,L))
927         *         /(A11**4.0)/48.0
928          DDAB22(LT,N,K,L)=((A22*DDB22(LT,N,K,L)+DA22(K,L)*DB22(LT,N)
929         *         -B22*DDA22(LT,N,K,L)-DB22(K,L)*DA22(LT,N))*(A22**2.0)
930         *         -(A22*DB22(LT,N)-B22*DA22(LT,N))*2*A22*DA22(K,L))
931         *         /(A22**4.0)/48.0
932    C
933          KP1(KT,LN)=BETA1*(AB11*DDAB11(LT,N,K,L)+DAB11(LT,N)*DAB11(K,L)
934         *                  +AB22*DDAB22(LT,N,K,L)+DAB22(LT,N)*DAB22(K,L))
935          KP11(KT,LN)=KP11(KT,LN)+KP1(KT,LN)
936    C
937          RETURN
938          END
939    C=============================================================
940    C     SUBROUTINE TO CALCULATE GEODISC MOMENT
941    C=============================================================
942          SUBROUTINE BEND2(AS,A11,A1,A2,AT211,AA11,TX211,TS211,LT,N,KT
943         *     ,BETA2,KP22,DA11,DA1,DA2,DAA11,DA12,DAS,DAT211,K,L,LN,A22
944         *     ,AA22,DTX11,DTS11,DDA11,DDA12,DDAS,TX122,TS122,DA22,DAA22
945         *     ,DTX22,DTS22,DAT122,AT122,A12,DDA22)
946    C
947          INTEGER LT,N,KT,K,L,LN
948          REAL A12,A11,A22,AS,AT211,AT122,BETA2,TX211,TS211,TX122,TS122
949         *    ,KP2(48,48),DTS22(16,3),DA12(16,3),DDA12(16,3,16,3)
950         *    ,DAS(16,3),DTS11(16,3),DDAS(16,3,16,3),KP22(48,48)
951         *    ,AA11(3),DA11(16,3),DAT211(16,3),DA1(16,3,3),DTX11(16,3)
952         *    ,AA22(3),DA22(16,3),DAT122(16,3),DA2(16,3,3),DTX22(16,3)
953         *    ,DDTS11(16,3,16,3),DDTX11(16,3,16,3),DAA11(16,3,3)
954         *    ,DDTS22(16,3,16,3),DDTX22(16,3,16,3),DAA22(16,3,3)
955         *    ,A1(3),DDA11(16,3,16,3),DDAT211(16,3,16,3)
956         *    ,A2(3),DDA22(16,3,16,3),DDAT122(16,3,16,3)
957    C
958          DDTX11(LT,N,K,L)=0.0
959          DDTX22(LT,N,K,L)=0.0
960          DO 50 M=1,3
961          DDTX11(LT,N,K,L)=DDTX11(LT,N,K,L)+(DA11(K,L)*DA2(LT,N,M)
962         *           +A2(M)*DDA11(LT,N,K,L)+DA2(K,L,M)*DA11(LT,N)
963         *           -DA12(K,L)*DA1(LT,N,M)-A1(M)*DDA12(LT,N,K,L)
964         *           -DA1(K,L,M)*DA12(LT,N))*AA11(M)
965         *           +(A11*DA2(LT,N,M)+A2(M)*DA11(LT,N)-A12*DA1(LT,N,M)
966         *           -A1(M)*DA12(LT,N))*DAA11(K,L,M)
967         *           +(A11*DA2(K,L,M)+DA11(K,L)*A2(M)-A12*DA1(K,L,M)
968         *           -DA12(K,L)*A1(M))*DAA11(LT,N,M)
969          DDTX22(LT,N,K,L)=DDTX22(LT,N,K,L)+(DA12(K,L)*DA2(LT,N,M)
970         *           +A2(M)*DDA12(LT,N,K,L)+DA2(K,L,M)*DA12(LT,N)
971         *           -DA22(K,L)*DA1(LT,N,M)-A1(M)*DDA22(LT,N,K,L)
972         *           -DA1(K,L,M)*DA22(LT,N))*AA22(M)
```

```
      *           +(A12*DA2(LT,N,M)+A2(M)*DA12(LT,N)-A22*DA1(LT,N,M)
      *           -A1(M)*DA22(LT,N))*DAA22(K,L,M)
      *           +(A12*DA2(K,L,M)+DA12(K,L)*A2(M)-A22*DA1(K,L,M)
      *           -DA22(K,L)*A1(M))*DAA22(LT,N,M)
 50    CONTINUE
       DDTS11(LT,N,K,L)=-((DA11(K,L)*DAS(LT,N)+DDAS(LT,N,K,L)*A11
      *                  +AS*DDA11(LT,N,K,L)+DAS(K,L)*DA11(LT,N))
      *                  *((AS*A11)**2.0)
      *                  -((A11*DAS(LT,N)+AS*DA11(LT,N))
      *                  *2.0*AS*A11*(DA11(K,L)*AS+A11*DAS(K,L))))
      *                  /((A11*AS)**4.0)/48.0
       DDTS22(LT,N,K,L)=-((DA22(K,L)*DAS(LT,N)+DDAS(LT,N,K,L)*A22
      *                  +AS*DDA22(LT,N,K,L)+DAS(K,L)*DA22(LT,N))
      *                  *((AS*A22)**2.0)
      *                  -((A22*DAS(LT,N)+AS*DA22(LT,N))
      *                  *2.0*AS*A22*(DA22(K,L)*AS+A22*DAS(K,L))))
      *                  /((A22*AS)**4.0)/48.0
       DDAT211(LT,N,K,L)=TX211*DDTS11(LT,N,K,L)+DTX11(K,L)*DTS11(LT,N)
      *                   +TS211*DDTX11(LT,N,K,L)+DTS11(K,L)*DTX11(LT,N)
       DDAT122(LT,N,K,L)=TX122*DDTS22(LT,N,K,L)+DTX22(K,L)*DTS22(LT,N)
      *                   +TS122*DDTX22(LT,N,K,L)+DTS22(K,L)*DTX22(LT,N)
C
       KP2(KT,LN)=BETA2*(AT211*DDAT211(LT,N,K,L)+DAT211(K,L)*
      *                  DAT211(LT,N)+AT122*DDAT122(LT,N,K,L)
      *                  +DAT122(K,L)*DAT122(LT,N))
       KP22(KT,LN)=KP22(KT,LN)+KP2(KT,LN)
C
        RETURN
        END
C=============================================================
C     SUBROUTINE TORQ
C=============================================================
       SUBROUTINE TORQ(AS,A12,AA12,B12,GAMA,KP33,DA12,DAS,A11,B11
      *    ,A22,B22,DNORU,DAA12,DB12,DDA12,DDAS,DDNORU,LT,N,KT,K,L,LN
      *    ,BS11,BX11,DB11,DDB11,DDA11,DABV11,DBX11,DBS11,DA11,ABV11
      *    ,BS22,BX22,DB22,DDB22,DDA22,DABV22,DBX22,DBS22,DA22,ABV22)
C
       INTEGER LT,N,KT,K,L,LN
       REAL A12,AS,GAMA,B12,AA12(3),KP3(48,48),DAS(16,3),KP33(48,48)
      *   ,DB12(16,3),DDB12(16,3,16,3),DDNORU(16,3,16,3,3),DAA12(16,3,3
      *   ,DDAS(16,3,16,3),DA12(16,3),DDA12(16,3,16,3),DNORU(16,3,3)
      *   ,A11,B11,ABV11,DB11(16,3),DDABV11(16,3,16,3),DDBS11(16,3,16,3
      *   ,A22,B22,ABV22,DB22(16,3),DDABV22(16,3,16,3),DDBS22(16,3,16,3
      *   ,DABV11(16,3),DDB11(16,3,16,3),DA11(16,3),DDA11(16,3,16,3)
      *   ,DABV22(16,3),DDB22(16,3,16,3),DA22(16,3),DDA22(16,3,16,3)
      *   ,BX11,BS11,DBX11(16,3),DBS11(16,3),DDBX11(16,3,16,3)
      *   ,BX22,BS22,DBX22(16,3),DBS22(16,3),DDBX22(16,3,16,3)
C
       DDB12(LT,N,K,L)=0.0
       DO 50 M=1,3
       DDB12(LT,N,K,L)=DDB12(LT,N,K,L)+(AA12(M)*DDNORU(LT,N,K,L,M)
      *    +DAA12(K,L,M)*DNORU(LT,N,M)+DAA12(LT,N,M)*DNORU(K,L,M))/48.0
 50    CONTINUE
       DDBX11(LT,N,K,L)=A12*DDB11(LT,N,K,L)+DA12(K,L)*DB11(LT,N)
```

```
1027        *                  +B11*DDA12(LT,N,K,L)+DA12(LT,N)*DB11(K,L)
1028        *                  -A11*DDB12(LT,N,K,L)-DA11(K,L)*DB12(LT,N)
1029        *                  -B12*DDA11(LT,N,K,L)-DA11(LT,N)*DB12(K,L)/48.0
1030         DDBX22(LT,N,K,L)=A22*DDB12(LT,N,K,L)+DA22(K,L)*DB12(LT,N)
1031        *                  +B12*DDA22(LT,N,K,L)+DA22(LT,N)*DB12(K,L)
1032        *                  -A12*DDB22(LT,N,K,L)-DA12(K,L)*DB22(LT,N)
1033        *                  -B22*DDA12(LT,N,K,L)-DA12(LT,N)*DB22(K,L)
1034         DDBS11(LT,N,K,L)=-((DA11(K,L)*DAS(LT,N)+DDAS(LT,N,K,L)*A11
1035        *                  +AS*DDA11(LT,N,K,L)+DAS(K,L)*DA11(LT,N))
1036        *                  *((AS*A11)**2.0)
1037        *                  -((A11*DAS(LT,N)+AS*DA11(LT,N))
1038        *                  *2*AS*A11*(DA11(K,L)*AS+A11*DAS(K,L))))
1039        *                  /((A11*AS)**4.0)
1040         DDBS22(LT,N,K,L)=-((DA22(K,L)*DAS(LT,N)+DDAS(LT,N,K,L)*A22
1041        *                  +AS*DDA22(LT,N,K,L)+DAS(K,L)*DA22(LT,N))
1042        *                  *((AS*A22)**2.0)
1043        *                  -((A22*DAS(LT,N)+AS*DA22(LT,N))
1044        *                  *2*AS*A22*(DA22(K,L)*AS+A22*DAS(K,L))))
1045        *                  /((A22*AS)**4.0)
1046         DDABV11(LT,N,K,L)=-(BX11*DDBS11(LT,N,K,L)+DBX11(K,L)*DBS11(LT,N)
1047        *                   +BS11*DDBX11(LT,N,K,L)+DBS11(K,L)*DBX11(LT,N))
1048         DDABV22(LT,N,K,L)=-(BX22*DDBS22(LT,N,K,L)+DBX22(K,L)*DBS22(LT,N)
1049        *                   +BS22*DDBX22(LT,N,K,L)+DBS22(K,L)*DBX22(LT,N))
1050    C
1051         KP3(KT,LN)=GAMA*(ABV11*DDABV11(LT,N,K,L)+DABV11(K,L)*DABV11(LT,N
1052        *                +ABV22*DDABV22(LT,N,K,L)+DABV22(K,L)*DABV22(LT,N)
1053         KP33(KT,LN)=KP33(KT,LN)+KP3(KT,LN)
1054    C
1055         RETURN
1056         END
1057    C=============================================================
1058    C      SUB. TO CALCULAT BENDING GLOBAL FORCE & STIFF.
1059    C=============================================================
1060         SUBROUTINE FBGKBG(IE,MCON,QP11,QP22,QP33,FB11,FB22,FB33
1061        *           ,KP11,KP22,KP33,KB11,KB22,KB33)
1062    C
1063         INTEGER  IE,MCON(64,16),IR(16)
1064         REAL  QP11(48),QP22(48),QP33(48),FB11(300),FB22(300),FB33(300)
1065        *     ,KP11(48,48),KP22(48,48),KP33(48,48),KB11(300,300)
1066        *     ,KB22(300,300),KB33(300,300)
1067    C
1068         DO 100 K=1,16
1069         IR(K)=MCON(IE,K)
1070    100  CONTINUE
1071    C
1072         DO 10 K=1,16
1073         DO 10 L=1,3
1074         IA=3*(IR(K)-1)+L
1075         JA=3*(K-1)+L
1076         FB11(IA)=FB11(IA)+QP11(JA)
1077         FB22(IA)=FB22(IA)+QP22(JA)
1078         FB33(IA)=FB33(IA)+QP33(JA)
1079    C
1080         DO 40 M=1,16
```

```
1081          DO 40 N=1,3
1082          IB=3*(IR(M)-1)+N
1083          JB=3*(M-1)+N
1084          KB11(IA,IB)=KB11(IA,IB)+KP11(JA,JB)
1085          KB22(IA,IB)=KB22(IA,IB)+KP22(JA,JB)
1086          KB33(IA,IB)=KB33(IA,IB)+KP33(JA,JB)
1087    40    CONTINUE
1088    10    CONTINUE
1089    C
1090          RETURN
1091          END
1092    C=============================================================
1093    C     SUBROUTINE TO CALCULATE THE TOTAL MATRICES
1094    C=============================================================
1095          SUBROUTINE GSTIF(FBG,FB11,FB22,FB33,KBG,KB11,KB22,KB33
1096         *           ,ROWN1,COLN1)
1097    C
1098          INTEGER   ROWN1,COLN1
1099          REAL FBG(300),FB11(300),FB22(300),FB33(300),BBG(300)
1100         *    ,KBG(300,300),KB11(300,300),KB22(300,300),KB33(300,300)
1101    C
1102          DO 63 K=1,ROWN1
1103          FBG(K)=FB11(K)+FB22(K)+FB33(K)
1104    C      BBG(K)=ABS(FBG(K))
1105    C      IF(BBG(K).LT.5) FBG(K)=0.0
1106          DO 63 L=1,COLN1
1107          KBG(K,L)=KB11(K,L)+KB22(K,L)+KB33(K,L)
1108    63    CONTINUE
1109    C
1110          RETURN
1111          END
1112    C=============================================================
1113    C     SUBROUTINE SPRNG  FORCES
1114    C=============================================================
1115          SUBROUTINE SPRNGF(NR,MCON,NRNO,ADA,SPRE,SPR,FS,DISP)
1116    C
1117          INTEGER  NR,NRNO,MCON(64,16),SPRE(50,2),IR(16)
1118          REAL SPR(50,4),ADA(4,4),H(1,4),U(1,4),V(4,1),VR(4,1)
1119         *    ,DISP(100,3,4,4),DE(100),DEL(100),FS(48),UA,UV,S,CONST
1120    C
1121          DO 10 I=1,48
1122          FS(I)=0.0
1123    10    CONTINUE
1124    C
1125          UA    =SPR(NR,1)
1126          VA    =SPR(NR,2)
1127          S     =SPR(NR,3)
1128          CONST=SPR(NR,4)
1129          U(1,1)=1.0
1130          U(1,2)=UA
1131          U(1,3)=UA**2.0
1132          U(1,4)=UA**3.0
1133          V(1,1)=1.0
1134          V(2,1)=VA
```

```
1135          V(3,1)=VA**2.0
1136          V(4,1)=VA**3.0
1137          DO 30 L=1,4
1138          H(1,L)=0.0
1139          VR(L,1)=0.0
1140          DO 30 K=1,4
1141          H(1,L)=H(1,L)+U(1,K)*ADA(K,L)
1142          VR(L,1)=VR(L,1)+ADA(K,L)*V(K,1)
1143    30    CONTINUE
1144    C
1145          IE=SPRE(NR,1)
1146          DE(NR)=0.0
1147          DO 35 I=1,4
1148          DO 35 J=1,4
1149          LT=4*(I-1)+J
1150          IF(SPRE(NR,2).EQ.2) GO TO 15
1151          IF(SPRE(NR,2).EQ.3) GO TO 16
1152          DE(NR)=DE(NR)+H(1,I)*DISP(IE,1,I,J)*VR(J,1)
1153          GO TO 35
1154    15    DE(NR)=DE(NR)+H(1,I)*DISP(IE,2,I,J)*VR(J,1)
1155          GO TO 35
1156    16    DE(NR)=DE(NR)+H(1,I)*DISP(IE,3,I,J)*VR(J,1)
1157    35    CONTINUE
1158          DEL(NR)=DE(NR)-CONST
1159    C      IF(ABS(DEL(NR)).LT.1E-6) DEL(NR)=0.0
1160    C
1161          DO 40 I=1,4
1162          DO 40 J=1,4
1163          LT=4*(I-1)+J
1164          IF(SPRE(NR,2).EQ.2) GO TO 17
1165          IF(SPRE(NR,2).EQ.3) GO TO 18
1166          JS=3*LT-2
1167          GO TO 180
1168    17    JS=3*LT-1
1169          GO TO 180
1170    18    JS=3*LT
1171    180   FS(JS)=S*DEL(NR)*H(1,I)*VR(J,1)
1172    40    CONTINUE
1173    C
1174          RETURN
1175          END
1176    C=============================================================
1177    C     SUB. SPRNGK FOR CALCULATING THE ELEMENT SPRING STIFF. M.
1178    C=============================================================
1179          SUBROUTINE SPRNGK(NR,MCON,NRNO,ADA,SPRE,SPR,KS)
1180    C
1181          INTEGER  NR,NRNO,MCON(64,16),SPRE(50,2),IR(16)
1182          REAL     SPR(50,4),ADA(4,4),U(1,4),H(1,4),KS(48,48)
1183         *        ,DE(100),DEL(100),UA,S,CONST,V(4,1),VR(4,1)
1184    C
1185          DO 10 I=1,48
1186          DO 10 J=1,48
1187          KS(I,J)=0.0
1188    10    CONTINUE
```

```
1189   C
1190         UA    =SPR(NR,1)
1191         VA    =SPR(NR,2)
1192         S     =SPR(NR,3)
1193         CONST=SPR(NR,4)
1194         U(1,1)=1.0
1195         U(1,2)=UA
1196         U(1,3)=UA**2.0
1197         U(1,4)=UA**3.0
1198         V(1,1)=1.0
1199         V(2,1)=VA
1200         V(3,1)=VA**2.0
1201         V(4,1)=VA**3.0
1202         DO 30 L=1,4
1203         H(1,L)=0.0
1204         VR(L,1)=0.0
1205         DO 30 K=1,4
1206         H(1,L)=H(1,L)+U(1,K)*ADA(K,L)
1207         VR(L,1)=VR(L,1)+ADA(K,L)*V(K,1)
1208   30    CONTINUE
1209   C
1210         DO 60 M=1,4
1211         DO 60 N=1,4
1212         LT=4*(M-1)+N
1213         IF(SPRE(NR,2).EQ.2) GO TO 36
1214         IF(SPRE(NR,2).EQ.3) GO TO 37
1215         MM=3*LT-2
1216         GO TO 370
1217   36    MM=3*LT-1
1218         GO TO 370
1219   37    MM=3*LT
1220   C
1221   370   DO 70 I=1,4
1222         DO 70 J=1,4
1223         KT=4*(I-1)+J
1224         IF(SPRE(NR,2).EQ.2) GO TO 46
1225         IF(SPRE(NR,2).EQ.3) GO TO 47
1226         NN=3*KT-2
1227         GO TO 470
1228   46    NN=3*KT-1
1229         GO TO 470
1230   47    NN=3*KT
1231   470   KS(MM,NN)=S*H(1,M)*VR(N,1)*H(1,I)*VR(J,1)
1232   70    CONTINUE
1233   60    CONTINUE
1234   C
1235         RETURN
1236         END
1237   C=================================================================
1238   C     SUB. TO CALCULAT THE SPRING GOLABL  FORCES & STIFF. MATRIX
1239   C=================================================================
1240         SUBROUTINE FSGKSG(NR,MCON,NRNO,ROWN1,COLN1,FS,FSG,KS,KSG,SPRE)
1241   C
1242         INTEGER  NR,NRNO,ROWN1,COLN1,IR(16),MCON(64,16),SPRE(50,2)
```

```
      REAL     FS(48),FSG(300),KS(48,48),KSG(300,300)
C
      MR=SPRE(NR,1)
      DO 10 K=1,16
      IR(K)=MCON(MR,K)
10    CONTINUE
C
      DO 31 I=1,16
      DO 31 M=1,3
      IA=3*(IR(I)-1)+M
      JA=3*(I-1)+M
      FSG(IA)=FSG(IA)+FS(JA)
C
      DO 50 J=1,16
      DO 50 N=1,3
      IB=3*(IR(J)-1)+N
      JB=3*(J-1)+N
      KSG(IA,IB)=KSG(IA,IB)+KS(JA,JB)
50    CONTINUE
31    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUBROUTINE V-COLUMN
C=============================================================
      SUBROUTINE VASUB(II,IENO,KZ,V,V1)
C
      INTEGER II,IENO,KZ
      REAL    VA,V(4,1),V1(4,1)
C
      VA=(II+KZ)/(2.0*IENO)
      V(1,1)=1.0
      V(2,1)=VA
      V(3,1)=VA**2.0
      V(4,1)=VA**3.0
C
      V1(2,1)=1.0
      V1(3,1)=2.0*VA
      V1(4,1)=3.0*(VA**2.0)
C
      RETURN
      END
C=============================================================
C     SUBROUTINE U-ROW
C=============================================================
      SUBROUTINE UASUB(JJ,IENO,JS,U,U1)
C
      INTEGER  JJ,IENO,JS
      REAL     U(1,4),U1(1,4),UA
C
      UA=(JJ+JS)/(2.0*IENO)
      U(1,1)=1.0
      U(1,2)=UA
```

```
      U(1,3)=UA**2.0
      U(1,4)=UA**3.0
C
      U1(1,2)=1.0
      U1(1,3)=2.0*UA
      U1(1,4)=3.0*(UA**2.0)
C
      RETURN
      END
C=============================================================
C     SUBROUTINE INITIAL VALUES
C=============================================================
      SUBROUTINE STRAIN(IE,ADA,DISP,U,U1,V,V1,II,JJ,AI11,AI22
     *                  ,A11,A22,STR,H,H1,VR,VR1,A1,A2,IENO)
C
      INTEGER   IENO,IE,II,JJ,IJ,MC,LF
      REAL U(1,4),U1(1,4),V(4,1),V1(4,1),C(1,4),C1(1,4),VR(4,1)
     *   ,VR1(4,1),DISP(100,3,4,4),ADA(4,4),H(1,4),H1(1,4)
     *   ,A11,A22,AI11,AI22,A1(3),A2(3),STR(8)
C
      DO 120 LF=1,4
      H(1,LF)=0.0
      H1(1,LF)=0.0
      VR(LF,1) =0.0
      VR1(LF,1)=0.0
      DO 120 K=1,4
      H(1,LF) =H(1,LF)+U(1,K)*ADA(K,LF)
      H1(1,LF)=H1(1,LF)+U1(1,K)*ADA(K,LF)
      VR(LF,1)=VR(LF,1)+ADA(K,LF)*V(K,1)
      VR1(LF,1)=VR1(LF,1)+ADA(K,LF)*V1(K,1)
120   CONTINUE
C
      A11=0.0
      A22=0.0
      DO 201 N=1,3
      A1(N)=0.0
      A2(N)=0.0
      DO 202 I=1,4
      C(1,I)=0.0
      C1(1,I)=0.0
      DO 202 J=1,4
      C(1,I) =C(1,I)+H(1,J)*DISP(IE,N,J,I)
      C1(1,I)=C1(1,I)+H1(1,J)*DISP(IE,N,J,I)
202   CONTINUE
      DO 203 I=1,4
      A1(N)=A1(N)+C(1,I)*VR1(I,1)
      A2(N)=A2(N)+C1(1,I)*VR(I,1)
203   CONTINUE
      A11=A11+A1(N)*A1(N)
      A22=A22+A2(N)*A2(N)
201   CONTINUE
C
      IJ=IENO*(II-1)+JJ
      DO 30 MC=1,2
```

```
1351          L=2*(IJ-1)+MC
1352          IF(MC.NE.1) GOTO 75
1353          STR(L)=(A11-AI11)/2.0*AI11
1354          GOTO 30
1355    C
1356    C75    STR(L)=(A22-AI22)/((A22/AI22)**0.5)
1357    C75    STR(L)=((A22-AI22)/(AI22))*(A22**0.5)
1358    75    STR(L)=(A22-AI22)/2.0*AI22
1359    30    CONTINUE
1360    C
1361          RETURN
1362          END
1363    C=============================================================
1364    C     SUBROUTINE AXIAL STIFFNES MATRIX
1365    C=============================================================
1366          SUBROUTINE AXIALS(IE,KQE,KRE,II,JJ,AI11,AI22,H,H1,VR,VR1,A1,A2
1367         *                  ,IENO,A11,A22)
1368    C
1369          INTEGER   IE,IENO,IJ,II,JJ,MC
1370          REAL H(1,4),H1(1,4),VR(4,1),VR1(4,1),AI11,AI22,A1(3),A2(3)
1371         * ,DA11(16,3),DA22(16,3),DA1(16,3,3),DA2(16,3,3)
1372         * ,KQE(8,48),KRE(48,8),A11,A22,KKQ(18,48)
1373    C
1374          IJ=IENO*(II-1)+JJ
1375          DO 30 MC=1,2
1376          L=2*(IJ-1)+MC
1377    C
1378          DO 100 I=1,4
1379          DO 100 J=1,4
1380          LT=4*(I-1)+J
1381          DO 100 N=1,3
1382          KT=3*(LT-1)+N
1383    C
1384          DA11(LT,N)=0.0
1385          DA22(LT,N)=0.0
1386          DO 301 M=1,3
1387          IF(M.NE.N) GO TO 302
1388          DA1(LT,N,M)=H(1,I)*VR1(J,1)
1389          DA2(LT,N,M)=H1(1,I)*VR(J,1)
1390          GO TO 301
1391    302   DA1(LT,N,M)=0.0
1392          DA2(LT,N,M)=0.0
1393    301   CONTINUE
1394    C
1395          DO 201 M=1,3
1396          DA11(LT,N)=DA11(LT,N)+(2.0*DA1(LT,N,M)*A1(M))
1397          DA22(LT,N)=DA22(LT,N)+(2.0*DA2(LT,N,M)*A2(M))
1398    201   CONTINUE
1399    C
1400          IF(MC.NE.1) GOTO 75
1401          KRE(KT,L)=-(DA11(LT,N)/2.0*AI11)
1402          KQE(L,KT)=-(DA11(LT,N)/2.0*AI11)
1403    C      KRE(KT,L)=KRE(KT,L)+((A11-AI11)/AI11)*(DA11(LT,N)/2.0*(A11**0.5
1404    C     *          +(A11**0.5)*(DA11(LT,N)/AI11)
```

```
1405   C       KQE(L,KT)=KQE(KT,L)+((A11-AI11)/AI11)*(DA11(LT,N)/2.0*(A11**0.5
1406   C      *          +(A11**0.5)*(DA11(LT,N)/AI11)
1407   C       KRE(KT,L)=KRE(KT,L)+DA11(LT,N)/(2.0*(A11**0.5))
1408   C       KQE(L,KT)=KQE(L,KT)+DA11(LT,N)/(2.0*(A11**0.5))
1409          KKQ(KT,L)=ABS(KRE(KT,L))
1410          IF(KKQ(KT,L).LT.1.0E-05) THEN
1411          KRE(KT,L)=0.0
1412          KQE(L,KT)=0.0
1413          ENDIF
1414          GOTO 100
1415   C
1416   75     KRE(KT,L)=-(DA22(LT,N)/2.0*AI22)
1417          KQE(L,KT)=-(DA22(LT,N)/2.0*AI22)
1418   C75     KRE(KT,L)=KRE(KT,L)+((A22-AI22)/AI22)*(DA22(LT,N)/2.0*(A22**0.5
1419   C      *          +(A22**0.5)*(DA22(LT,N)/AI22)
1420   C       KQE(L,KT)=KQE(KT,L)+((A22-AI22)/AI22)*(DA22(LT,N)/2.0*(A22**0.5
1421   C      *          +(A22**0.5)*(DA22(LT,N)/AI22)
1422   C75     KRE(KT,L)=KRE(KT,L)+DA22(LT,N)/(2.0*(A22**0.5))
1423   C       KQE(L,KT)=KQE(L,KT)+DA22(LT,N)/(2.0*(A22**0.5))
1424          KKQ(KT,L)=ABS(KRE(KT,L))
1425          IF(KKQ(KT,L).LT.1.0E-05) THEN
1426          KRE(KT,L)=0.0
1427          KQE(L,KT)=0.0
1428          ENDIF
1429   100    CONTINUE
1430   30     CONTINUE
1431   C
1432          RETURN
1433          END
1434   C=============================================================
1435   C      SUB. TO CALCULAT THE GLOBAL AXIAL FORCES & STIFF. MATRIX
1436   C=============================================================
1437          SUBROUTINE KAGMAT(IE,IENO,MCON,KRE,KRG,KQE,KQG,STR,STRG)
1438   C
1439          INTEGER  IE,IENO,MCON(64,16),IR(16)
1440          REAL KQE(8,48),KQG(200,300),KRE(48,8),KRG(300,200)
1441         *      ,STR(8),STRG(200)
1442   C
1443          DO 10 K=1,16
1444          IR(K)=MCON(IE,K)
1445   10     CONTINUE
1446   C
1447          MV=2*(IENO**2)
1448          DO 100 LN=1,MV
1449          NN=MV*(IE-1)+LN
1450          STRG(NN)=0.0
1451          STRG(NN)=STRG(NN)+STR(LN)
1452          DO 100 K=1,16
1453          DO 100 L=1,3
1454          IA=3*(IR(K)-1)+L
1455          JA=3*(K-1)+L
1456          KRG(IA,NN)=0.0
1457          KQG(NN,IA)=0.0
1458          KRG(IA,NN)=KRG(IA,NN)+KRE(JA,LN)
```

```
1459          KQG(NN,IA)=KQG(NN,IA)+KQE(LN,JA)
1460    100   CONTINUE
1461    C
1462          RETURN
1463          END
1464    C=============================================================
1465    C     SUB. TO CALCULAT THE TOTAL FORCE AND STIFFNES MATRIX
1466    C=============================================================
1467          SUBROUTINE PGKGMT(FBG,FSG,FPG,PG,KBG,KRG,KQG,KSG,KG,ROWN,COLN
1468         *            ,ROWN1,COLN1,ROWNA,COLNA,KGG,PGG,STRG,TENG,PT)
1469    C
1470          INTEGER ROWN,COLN,ROWN1,COLN1,ROWNA,COLNA
1471    C
1472          REAL KBG(300,300),KRG(300,200),KQG(200,300),TENG(200),FAG(300)
1473         *  ,FBG(300),FPG(300),FSG(300),PG(300),PGG(300),STRG(200)
1474         *  ,PT(300),KG(300,300),KGG(300,300),KSG(300,300),BBG(300)
1475    C
1476          DO 40 I=1,ROWN1
1477          DO 40 J=1,COLN1
1478          KG(I,J)=KBG(I,J)+KSG(I,J)
1479    40    CONTINUE
1480          DO 50 M=1,ROWN1
1481          DO 50 N=1,COLNA
1482          KG(M,COLN1+N)=KRG(M,N)
1483          KG(ROWN1+N,M)=KQG(N,M)
1484    50    CONTINUE
1485    C
1486    C      DO 60 I=1,ROWN
1487    C      DO 60 J=1,COLN
1488    C      KGG(I,J)=KG(I,J)
1489    C60    CONTINUE
1490    C
1491          DO 100 I=1,ROWN1
1492          FAG(I)=0.0
1493          DO 100 J=COLNA
1494          FAG(I)=FAG(I)+KRG(I,J)*TENG(J)
1495    100   CONTINUE
1496    C
1497          DO 10 I=1,ROWN1
1498          PG(I)=FPG(I)-(FBG(I)+FSG(I)+FAG(I))
1499          BBG(I)=ABS(PG(I))
1500          IF(BBG(I).LT.5E-4) PG(I)=0.0
1501    10    CONTINUE
1502          DO 20 J=1,ROWNA
1503          JJ=ROWN1+J
1504          PG(JJ)=STRG(J)
1505    20    CONTINUE
1506    C
1507          RETURN
1508          END
1509    C=============================================================
1510    C     SUB. SOLVE BY USING GAUAS.ELEMENATION.
1511    C=============================================================
1512          SUBROUTINE    SOLVE(NX,MX,KG,BY,DY,PCC)
```

```
1513   C
1514         INTEGER     NX,MX
1515         REAL        BY(300),DY(300),KG(300,300),PCC(300)
1516   C
1517         M1=NX
1518         M2=M1-1
1519         DO 300 I=1,M2
1520         II=I+1
1521         DO 290 K=II,M1
1522         FACT=KG(K,I)/KG(I,I)
1523         DO 280 J=II,M1
1524         KG(K,J)=KG(K,J)-FACT*KG(I,J)
1525   280   CONTINUE
1526         KG(K,I)=0.0
1527         BY(K)=BY(K)-FACT*BY(I)
1528   290   CONTINUE
1529   300   CONTINUE
1530   C
1531   C      BACK SUBSTITUE
1532   C
1533         DO 400 I=1,M1
1534         II=M1-I+1
1535         PIVOT=KG(II,II)
1536         KG(II,II)=0.0
1537         DO 350 J=II,M1
1538         BY(II)=BY(II)-KG(II,J)*DY(J)
1539   350   CONTINUE
1540         DY(II)=BY(II)/PIVOT
1541   400   CONTINUE
1542   C
1543         RETURN
1544         END
1545   C=============================================================
1546   C      SUB. SOLVE BY USING NAG-SUB. LIB.
1547   C=============================================================
1548        SUBROUTINE     SOLVEN(NX,MX,KG,BY,DY,PCC,DCC)
1549   C
1550         INTEGER             NX,MX,I,NN,J,IA,IAA,IFAIL
1551         REAL DCC(300),PCC(300),KG(300,300),BY(300),DY(300),KKG(300,300)
1552        *    ,DDY(300)
1553         DOUBLE PRECISION  A1(300,300),BB(300),CC(300),AA1(300,300)
1554        *                 ,WKS1(300),WKS2(300)
1555   C
1556         NN=MX
1557         IFAIL=1
1558   C
1559         DO 40 I=1,NX
1560         BB(I)=BY(I)
1561         DO 40 J=1,MX
1562         A1(I,J)=KG(I,J)
1563   40    CONTINUE
1564   C
1565         CALL F04ATF(A1,300,BB,NN,CC,AA1,300,WKS1,WKS2,IFAIL)
1566         IF(IFAIL.EQ.0) GOTO 20
```

```
      WRITE(6,9998)IFAIL
9998  FORMAT('IFAIL= ',I2)
      STOP
C
20    DO 50 I=1,NX
      DY(I)=CC(I)
      DO 50 J=1,MX
      KKG(I,J)=AA1(I,J)
50    CONTINUE
C
      DO 30 I=1,NX
      PCC(I)=0.0
      DDY(I)=0.0
      DO 60 J=1,MX
      PCC(I)=PCC(I)+KG(I,J)*DY(J)
60    CONTINUE
30    CONTINUE
C
      RETURN
      END
C=============================================================
C     SUB. OUTPUT THE NODE DISPLACEMENT and ELEM. TENSION
C=============================================================
      SUBROUTINE OUTSO1(NE,NOCP,IENO,ROWN1,DG,X,Y,Z,TX,TY,TENG
     *                 ,MCON,CORD,DISP)
C
      INTEGER NE,NOCP,IENO,NOSE,ROWN1,MCON(64,16)
      REAL    DG(300),X(121),Y(121),Z(121),TX(200),TY(200),TENG(200)
     *       ,CORD(121,3),DISP(100,3,4,4)
C
C      WRITE(25,*)('  NO      D(1)        D(2)        D(3) ')
      DO 50 I=1,NOCP
      ID=3*I-2
      X(I)=X(I)+DG(ID)
      Y(I)=Y(I)+DG(ID+1)
      Z(I)=Z(I)+DG(ID+2)
      CORD(I,1)=X(I)
      CORD(I,2)=Y(I)
      CORD(I,3)=Z(I)
      WRITE(25,4050)I,(CORD(I,M),M=1,3)
50    CONTINUE
4050  FORMAT(I10,3E14.4)
C
      DO 400 IE=1,NE
      DO 400 I=1,4
      DO 400 J=1,4
      LT=4*(I-1)+J
      L=MCON(IE,LT)
      DISP(IE,1,I,J)=CORD(L,1)
      DISP(IE,2,I,J)=CORD(L,2)
      DISP(IE,3,I,J)=CORD(L,3)
400   CONTINUE
C
      NOSE=NE*(IENO**2)
```

```
1621  C       WRITE(25,*)('ELEMENT TENSIONS,')
1622  C       WRITE(25,*)('I        TU(I)        TV(I)')
1623          DO 551 I=1,NOSE
1624          TENG(I)=DG(ROWN1+I)
1625          IX=ROWN1+(2*I)-1
1626          IY=ROWN1+(2*I)
1627          TX(I)=TX(I)+DG(IX)
1628          TY(I)=TY(I)+DG(IY)
1629  C       WRITE(25,4055)I,TX(I),TY(I)
1630  C4055   FORMAT(I4,2E15.3)
1631  551     CONTINUE
1632  C
1633          RETURN
1634          END
1635  C=============================================================
1636  C       SUB. OUTPUT THE NODE DISPLACEMENT
1637  C=============================================================
1638          SUBROUTINE OUTSO2(NE,MCON,NOCP,DG,DOF,X,Y,Z,RX,RY,RZ,ADA,CORD)
1639  C
1640          INTEGER  NE,NC,NPE,TNE,NOCP,CNE,NN,MCON(64,16),N(16)
1641          REAL  CORD(121,3),X(121),Y(121),Z(121),ADA(4,4)
1642         *      ,TMX(4,4),TMY(4,4),TMZ(4,4),U,V,RX,RY,RZ
1643         *      ,AMX(4,4),AMY(4,4),AMZ(4,4),UM(1,4),VM(4,1)
1644         *,FMX(1,4),FMY(1,4),FMZ(1,4),AMXX(4,4),AMYY(4,4),AMZZ(4,4)
1645  C
1646          DO 80 IE=1,NE
1647          WRITE(25,29)IE
1648  29      FORMAT('ELEM. NO.  ',I5)
1649          DO 13 I=1,16
1650          N(I)=MCON(IE,I)
1651  13      CONTINUE
1652          MY=0
1653          DO 14 I=1,4
1654          DO 15 J=1,4
1655          L=I+J-1+MY
1656          TMX(I,J)=X(N(L))
1657          TMY(I,J)=Y(N(L))
1658          TMZ(I,J)=Z(N(L))
1659  15      CONTINUE
1660          MY=MY+3
1661  14      CONTINUE
1662  C
1663          DO 17 I=1,4
1664          DO 17 J=1,4
1665          AMXX(I,J)=0.0
1666          AMYY(I,J)=0.0
1667          AMZZ(I,J)=0.0
1668          DO 17 M=1,4
1669          AMXX(I,J)=AMXX(I,J)+ADA(I,M)*TMX(M,J)
1670          AMYY(I,J)=AMYY(I,J)+ADA(I,M)*TMY(M,J)
1671          AMZZ(I,J)=AMZZ(I,J)+ADA(I,M)*TMZ(M,J)
1672  17      CONTINUE
1673  C
1674          DO 93 II=1,4
```

```
1675          DO 93 JJ=1,4
1676          AMX(II,JJ)=0.0
1677          AMY(II,JJ)=0.0
1678          AMZ(II,JJ)=0.0
1679          DO 93 MM=1,4
1680          AMX(II,JJ)=AMX(II,JJ)+AMXX(II,MM)*ADA(JJ,MM)
1681          AMY(II,JJ)=AMY(II,JJ)+AMYY(II,MM)*ADA(JJ,MM)
1682          AMZ(II,JJ)=AMZ(II,JJ)+AMZZ(II,MM)*ADA(JJ,MM)
1683    93    CONTINUE
1684    C
1685          INN=0
1686          DO 70 JU=1,4
1687          U=(JU-1)/3.0
1688          UM(1,1)=1.0
1689          UM(1,2)=U
1690          UM(1,3)=(U**2)
1691          UM(1,4)=(U**3)
1692    C
1693          DO 60 JV=1,4
1694          V=(JV-1)/3.0
1695          VM(1,1)=1.0
1696          VM(2,1)=V
1697          VM(3,1)=(V**2)
1698          VM(4,1)=(V**3)
1699    C
1700          DO 10 J=1,4
1701          FMX(1,J)=0.0
1702          FMY(1,J)=0.0
1703          FMZ(1,J)=0.0
1704          DO 10 M=1,4
1705          FMX(1,J)=FMX(1,J)+UM(1,M)*AMX(M,J)
1706          FMY(1,J)=FMY(1,J)+UM(1,M)*AMY(M,J)
1707          FMZ(1,J)=FMZ(1,J)+UM(1,M)*AMZ(M,J)
1708    10    CONTINUE
1709    C
1710          RX=0.0
1711          RZ=0.0
1712          RY=0.0
1713          DO 20 J=1,4
1714          RX=RX+FMX(1,J)*VM(J,1)
1715          RY=RY+FMY(1,J)*VM(J,1)
1716          RZ=RZ+FMZ(1,J)*VM(J,1)
1717    20    CONTINUE
1718    C
1719          WRITE(25,19)RX,RY,RZ
1720    19    FORMAT(3E14.4)
1721    C
1722    60    CONTINUE
1723    70    CONTINUE
1724    80    CONTINUE
1725    C
1726          RETURN
1727          END
1728    C=============================================================
```

The buckling analysis of grid shells using bicubic B-spline finite elements

C.J.K. Williams, M.A.,
School of Architecture & Building Engineering, University of Bath

and

M.A. Mohamedien, B.Sc., M.Sc.,
Department of Civil Engineering, Suez Canal University, Port-Said, Egypt.

Synopsis

Bicubic B-spline surface 'patches' are used for the specification of surfaces in computer aided geometric design for purposes such as the control of machine tools. This paper describes the theory of the application of such patches as finite elements for the buckling analysis of grid shells which may undergo large deflections prior to collapse. Good agreement was found between results from the numerical analysis and tests on a physical model.

Introduction

The commonest example of a 'grid' or 'lattice' shell is a dome shaped kitchen sieve made of woven wire mesh. The mesh is fabricated flat and it is possible to form a dome shape since there can be relative rotation of the two sets of wires so that the wires form little diamond shapes on the surface instead of squares.

The concept of the grid shell was first suggested by the architect, Professor Frei Otto[1], together with the idea of forming the shell in the shape of a hanging chain net which is then inverted. This has the advantage that there will be no bending moments in the members due to own weight.

The Mannheim grid shells described by Happold and Liddell[2] and Otto[3] are made of continuous timber members which are not woven, but joined by single bolts where they cross to allow relative rotation of the two sets of members. The shells are of irregular shape and were fabricated flat and then erected on scaffold towers which were lifted using fork-lift trucks.

The Mannheim shells span up to 60m and have two layers of timber members running in each direction, making four layers altogether. This is to provide sufficient bending stiffness, but nevertheless the total thickness of the shells is only 200mm.

The relative rotation of the members which is so essential to the erection of grid shells also allows mechanisms of deformation of the shell after erection. This raises the question of failure of the shells due to buckling, particularly under snow load. During the design of the Mannheim shells it was found necessary to introduce diagonal wire ties to stiffen the structure.

However, the flexibility caused by the relative rotation of the two sets of members does have the advantage that deflections prior to collapse are relatively large which in turn means that grid shells show more warning of buckling collapse than do conventional shells. It also means that the buckling load is not greatly effected by small geometrical imperfections, unlike conventional shells.

There seems to be no analytical method which can be used to predict the buckling load of grid shells due to their highly non-linear behaviour. This means that one is forced to rely on physical model tests and numerical methods. However sophisticated the numerical model, it is the authors' view that physical model tests are still indispensable in the design of grid shells. This is because the modes of deformation of grid shells are so complex that only a physical model which can be prodded and poked under load gives a true feeling of structural action. One of the authors (Williams) was involved in the numerical analysis and physical model testing of the Mannheim shells and it was found that when a physical model was near to collapse certain areas became very stiff, pushing against the ties, while other regions lost all stiffness and were hanging in tension off the stiff areas.

Surface coordinates

A grid shell consists of members running in two directions. One could, perhaps, produce a structural model which models each member on a one to one

basis. However, for a reasonably fine grid this would be very uneconomic in computing resources. One possibility is to try and model the contribution of a number of parallel members in one 'beam element' as was done on the Mannheim shells. The alternative is to say that a fine grid is nearer to a continuous surface than it is to a coarse grid. This is the approach that we have adopted.

The most convenient way to specify a surface is using the parametric form

$$\begin{aligned} x &= x(u, v) \\ y &= y(u, v) \\ z &= z(u, v). \end{aligned} \qquad 1$$

u and v are known as surface coordinates. Any one surface can be expressed in an infinite number of different parametric forms. For example, the hyperboloid of one sheet,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1 \qquad 2$$

could be written as

$$\begin{aligned} x &= a \cosh u \cos v \\ y &= b \cosh u \sin v \\ z &= c \sinh u \end{aligned} \qquad 3$$

or, with different surface coordiates, as

$$\begin{aligned} x &= a\left[\frac{\cos(u + v)}{\cos(u - v)}\right] \\ y &= b\left[\frac{\sin(u + v)}{\cos(u - v)}\right] \\ z &= c \tan(u - v). \end{aligned} \qquad 4$$

Eliminating u and v from the three equations, 3, or the three equations, 4, gives equation 2 in each case.

Lines of constant u and constant v form a net on a surface in the same way that lines of latitude and longitude form a net on a globe.

In order to avoid having to write separate equations for x, y and z, it is convenient to introduce the position vector,

$$\mathbf{r} = \mathbf{r}(u, v) = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \qquad 5$$

in which $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are unit vectors in the directions of x, y and z respectively.

It would seem natural to try and find one vector function, $\mathbf{r}(u, v)$, to repre-

sent an entire surface. However, this leads to a number of problems and it is easier to represent a surface by a number of elements or 'patches'. Each element has its own surface coordinate system (u and v) and its own vector function $\mathbf{r}(u, v)$ to describe its shape.

We shall be dividing the surface of a grid shell up into curved quadrilateral shaped elements and the four sides of each element will be the curved lines $u = 0$ and 1 and $v = 0$ and 1. Along the common boundary between elements 1 and 2 shown in figure 1, $u = 1$ on element 1 and $u = 0$ on element 2, but at a particular point on the boundary both elements share the same value of v. On the boundary between elements 1 and 3 u is continuous and v jumps back from 1 to 0.

The members of a grid shell form an obvious system of surface coordinates and therefore we shall choose the elements and their surface coordinates so that either u = constant or v = constant along the members.

Even though we can allow jumps in the coordinate system between elements, we must ensure that there is sufficient 'physical' continuity between elements. No overlaps or gaps can be allowed to form between elements and we must also make sure that no 'kinks' occur in the continuous members of a grid shell as they pass from pass element to another.

Much of the work in the application of the finite element method to plates and shells concentrates on producing elements which provide sufficient continuity across boundaries. The vector function $\mathbf{r}(u, v)$ for each element is specified in terms of 'degrees of freedom' which are the coordinates of corner 'nodes', the directions or slopes of certain tangents and so on. Adjactent elements share certain degrees of freedom and this provides the continuity. The more degrees of freedom that adjacent elements share, the better the continuity.

The bicubic B-spline surface patch

The bicubic B-spline surface patch (or element, we shall use either word as seems most appropriate) is used for the specification of surfaces in computer aided geometric design for purposes such as the control of machine tools. In this application there is again the necessity for continuity of position and tangent between patches. In fact the bicubic B-spline also provides continuity of curvature between adjacent patches, although in our application this is not strictly necessary.

The bicubic B-spline patch achieves this level of continuity from a relatively simple formulation by ensuring that adjacent patches share 3/4 of their degrees of freedom. Not only that, but the bicubic B-spline patch only has coordinates for its degrees of freedom, not rotations or the orientation of tangents - this makes it particularly attractive for large deflection problems such as grid shells.

The theory of computer aided geometric design and the bicubic B-spline patch is described in detail in Faux and Pratt[4]. The function $\mathbf{r}(u, v)$ specifying the shape and position of a bicubic B-spline patch can be written

$$\mathbf{r} = \mathbf{r}(u, v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \mathbf{D} \mathbf{P} \mathbf{D}^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \qquad 6$$

$$0 \le u \le 1$$
$$0 \le v \le 1$$

in which

$$\mathbf{D} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \qquad 7$$

and

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_{11} & \mathbf{p}_{12} & \mathbf{p}_{13} & \mathbf{p}_{14} \\ \mathbf{p}_{21} & \mathbf{p}_{22} & \mathbf{p}_{23} & \mathbf{p}_{24} \\ \mathbf{p}_{31} & \mathbf{p}_{32} & \mathbf{p}_{33} & \mathbf{p}_{34} \\ \mathbf{p}_{41} & \mathbf{p}_{42} & \mathbf{p}_{43} & \mathbf{p}_{44} \end{bmatrix}. \qquad 8$$

The matrix $\mathbf{P}$ contains 16 vectors, $\mathbf{p}_{11}$ to $\mathbf{p}_{44}$, which are the position vectors of 16 'control points' relative to the origin of x, y and z coordinates. $\mathbf{r}$ is calculated using the usual rules of matrix multiplication - 'rows into columns'. The vector equation, equation 6, is equivalent to 3 ordinary equations. Thus, if we were to put the x coordinates of the 16 control points, x_{11} to x_{44}, into $\mathbf{P}$ instead of $\mathbf{p}_{11}$ to $\mathbf{p}_{44}$, equation 6 would give $x(u, v)$, the x coordinate of a typical point on the surface.

The adjective 'bicubic' comes from the fact that equation 6 contains u and v up to the cubic term.

In general, none of the 16 control points actually lies on the surface, which seems a little odd at first, but it is necessary to the 'working' of the patch. The con-

trol points for two adjacent patches overlap as shown in figure 2 on which it can be seen that the two patches share 12 control points.

If patch 1 is specified by $\mathbf{P}$ as given in equation 8, patch 2 would be specified by

$$\mathbf{P} = \begin{bmatrix} P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \\ * & * & * & * \end{bmatrix} \qquad 9$$

in which the * denote the position vectors of the four unlabled control points on the right hand side of figure 2.

Now substituting $u = 1$ into equation 6 for patch 1 (in which $\mathbf{P}$ is obtained from equation 8) and $u = 0$ into equation 6 for patch 2 (in which $\mathbf{P}$ is obtained from equation 9) gives the same result for $\mathbf{r}$ as a function of v. Thus there is continuity of position between the two patches.

There is also continuity of the partial derivatives

$$\frac{\partial \mathbf{r}}{\partial u} = \begin{bmatrix} 0 & 1 & 2u & 3u^2 \end{bmatrix} \mathbf{D}\mathbf{P}\mathbf{D}^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \qquad 10$$

and

$$\frac{\partial^2 \mathbf{r}}{\partial u^2} = \begin{bmatrix} 0 & 0 & 2 & 6u \end{bmatrix} \mathbf{D}\mathbf{P}\mathbf{D}^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \qquad 11$$

and therefore there is continuity of slope and curvature.

There is similar continuity across boundaries across which v jumps back from 1 to 0.

Faux and Pratt[4] describe how the formulation, equation 6, is obtained to achieve this apparently miraculous result, but it is enough for us to understand how to use the bicubic B-spline patch and, if desired, check that it does work using the above reasoning. The factor 1/6 in equation 7 is necessary so that if all the control points were to lie in a plane, then the patch would lie in the same plane.

We have chosen to use a bicubic B-spline patch or finite element, but it would be possible to use a biquadratic B-spline instead. This would have $3 \times 3 = 9$

control points instead of 16. The biquadratic patch would give continuity of slope (which is all that we strictly require), but not continuity of curvature. However, the biquadratic finite element should not be dismissed since it is often better to use a larger number of simpler elements.

The fact that the control points do not in general lie on the surface does not prevent us using their coordinates as degrees of freedom in a finite element analysis. One might argue that it is impossible to apply loads to nodes that are not on the surface, but, of course, a proper finite element analysis only applies 'fictitious' or 'generalised' loads to nodes. These fictitious loads are obtained from the real loads by integration using the virtual work equation.

Virtual work equation

In the application of the finite element method to structural problems the equations of equilibrium are built up using the virtual work equation or one of the strain energy theorems which amount to the same thing. For our purposes the virtual work equation can be written:

$$\begin{aligned}
&\iint [\mathbf{q}.\delta\mathbf{r}]\, du\, dv \\
-&\iint [M_{uu}\,\delta\kappa_{uu} + M_{vv}\,\delta\kappa_{vv} + M_{uv}\,\delta\kappa_{uv} + T_u\,\delta\varepsilon_u]\, du\, dv \\
-&\iint [M_{uv}\,\delta\kappa_{uv} + M_{vv}\,\delta\kappa_{vv} + M_{uv}\,\delta\kappa_{uv} + T_v\,\delta\varepsilon_v]\, du\, dv \\
&-\iint [T_{uv}\,\delta\varepsilon_{uv}]\, du\, dv \\
&-\int_{Boundary} [\mathbf{Q}.\delta\mathbf{r} + \mathbf{M}.\delta\boldsymbol{\Omega}]\, ds = 0 \qquad 12
\end{aligned}$$

in which the 'equilibrium set' and the 'compatibility set' are

Equilibrium set

$\mathbf{q}$ = load on shell per unit α times γ

$M_{\alpha\alpha}$ (or $M_{\beta\gamma}$) = normal bending moment in members in the direction of increasing α (or γ)

$M_{\beta\alpha}$ (or $M_{\beta\gamma}$) = geodesic bending moment in members in the direction of increasing α (or γ)

$M_{\gamma\alpha}$ (or $M_{\gamma\gamma}$) = torsional moment in members in the direction of increasing α (or γ)

T_α (or T_γ) = tension in members in the direction of increasing α (or γ)

$T_{\alpha\alpha}$ = tension in diagonal ties

$\mathbf{Q}$ = forces exerted by shell on boundary per unit length

$\mathbf{M}$ = moments exerted by shell on boundary per unit length

and

Compatibility set

$\delta\mathbf{r}$ = increment of displacement of shell

$\delta\boldsymbol{\Omega}$ = increment of rotation of shell at boundary

$\delta\kappa_{\alpha\alpha}$ (or $\delta\kappa_{\alpha\gamma}$) = increment of normal curvature in members in the direction of increasing α (or γ)

$\delta\kappa_{\beta\alpha}$ (or $\delta\kappa_{\beta\gamma}$) = increment of geodesic curvature in members in the direction of increasing α (or γ)

$\delta\kappa_{\gamma\alpha}$ (or $\delta\kappa_{\gamma\gamma}$) = increment of twist per unit length in members in the direction of increasing α (or γ)

$\delta\varepsilon_\alpha$ (or $\delta\varepsilon_\gamma$) = increment of axial strain in members in the direction of increasing α (or γ)

$\delta\varepsilon_{\alpha\alpha}$ = increment of strain in diagonal ties.

s is arc length along the boundary.

The detailed definitions of the quantities in equation 12 must be such that when each member of the equilibrium set is multiplied by the appropriate member of the compatibility set and the result integrated over the surface or boundary, then the result is equal to the virtual work that is done or absorbed. Thus certain quanti-

ties need to be defined as 'per unit u' or 'per unit v'. The words 'normal' and 'geodesic' will be defined below.

The moments, shear forces and tensions (or, more likely, compressions, i.e. negative tensions) in the grid members and the tensions in any ties are in equilibrium with the applied loads, $\mathbf{q}$, and the boundary forces and moments. The increments of curvature, twist and axial strain are compatible with the increment of displacement, $\delta\mathbf{r}$, of a typical point on the surface or boundary and the increment of rotation, $\delta\Omega$, which occurs at the boundary.

In our application of the virtual work equation the equilibrium set will be 'real' and the compatibility set will be 'hypothetical'. Thus a number of differing hypothetical compatibility sets will be used to build up the 'equilibrium equations' following the standard procedure of the finite element method.

It is important to realise that we can use the virtual work equation in a non-linear, large displacement problem provided that we use a compatibility set which consists of small increments of displacement, rotation, curvature, twist and axial strain which take place about the deformed geometry.

Shear forces do not appear in the virtual work equation, except at the boundary where they are included in $\mathbf{Q}$. On the Mannheim shells composite action between the two layers of grid running in the same direction was not perfect so that it was necessary to consider deformation caused by shear force. This effect could be included in the method now being discussed by including the relative 'slip' between the two parallel layers of grid as extra degrees of freedom and introducing the appropriate extra terms in the virtual work equation.

In order to make practical use of the virtual work equation, equation 12, we need to be able to calculate the curvature, twist and axial strain of the grid shell members and also the increments of these quantities associated with an increment of displacement.

The unit tangent to the grid members running in the direction of increasing u is

$$\mathbf{t}_u = \frac{\dfrac{\partial \mathbf{r}}{\partial u}}{\left|\dfrac{\partial \mathbf{r}}{\partial u}\right|} = \frac{\dfrac{\partial \mathbf{r}}{\partial u}}{\sqrt{\dfrac{\partial \mathbf{r}}{\partial u}\cdot\dfrac{\partial \mathbf{r}}{\partial u}}} \qquad 13$$

and the unit normal to the surface,

$$\mathbf{n} = \frac{\frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v}}{\left| \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right|}. \qquad 14$$

The curvature of the grid members can be divided into two parts, normal curvature and geodesic curvature. The normal curvature is the curvature lying in a plane normal to the surface. The normal curvature in the direction of increasing u,

$$\kappa_{nu} = \frac{\frac{\partial \mathbf{t}_u}{\partial u} . \mathbf{n}}{\left| \frac{\partial \mathbf{r}}{\partial u} \right|} = \frac{\frac{\partial^2 \mathbf{r}}{\partial u^2} . \mathbf{n}}{\frac{\partial \mathbf{r}}{\partial u} . \frac{\partial \mathbf{r}}{\partial u}}. \qquad 15$$

The geodesic curvature is the curvature lying in the plane of the surface, that is the curvature seen when looking in a direction perpendicular to the surface. A 'geodesic' is a line of zero geodesic curvature on a surface. The geodesic curvature of a member running in the direction of increasing u is

$$\kappa_{gu} = \frac{\frac{\partial \mathbf{t}_u}{\partial u} . [\mathbf{n} \times \mathbf{t}_u]}{\left| \frac{\partial \mathbf{r}}{\partial u} \right|} = \frac{\frac{\partial^2 \mathbf{r}}{\partial u^2} . \left[\mathbf{n} \times \frac{\partial \mathbf{r}}{\partial u} \right]}{\left[\frac{\partial \mathbf{r}}{\partial u} . \frac{\partial \mathbf{r}}{\partial u} \right]^{\frac{3}{2}}}. \qquad 16$$

The twist per unit length of a member in the direction of increasing u is

$$\kappa_{tu} = \frac{\frac{\partial \mathbf{n}}{\partial u} . [\mathbf{n} \times \mathbf{t}_u]}{\left| \frac{\partial \mathbf{r}}{\partial u} \right|}$$

$$= \frac{\left[\frac{\partial^2 \mathbf{r}}{\partial u^2} \times \frac{\partial \mathbf{r}}{\partial v} + \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial^2 \mathbf{r}}{\partial u \partial v} \right] . \left[\mathbf{n} \times \frac{\partial \mathbf{r}}{\partial u} \right]}{\left[\frac{\partial \mathbf{r}}{\partial u} . \frac{\partial \mathbf{r}}{\partial u} \right] \left| \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right|}. \qquad 17$$

The axial strain in the grid members in the direction of increasing u is

$$\varepsilon_u = \frac{1}{2}\left[\frac{\frac{\partial \mathbf{r}}{\partial u}\cdot\frac{\partial \mathbf{r}}{\partial u}}{L_u^2} - 1\right] \qquad 18$$

$$= \frac{1}{2}\left[\frac{\sqrt{\frac{\partial \mathbf{r}}{\partial u}\cdot\frac{\partial \mathbf{r}}{\partial u}}}{L_u} + 1\right]\left[\frac{\sqrt{\frac{\partial \mathbf{r}}{\partial u}\cdot\frac{\partial \mathbf{r}}{\partial u}}}{L_u} - 1\right]$$

in which

$$L_u^2 = \left[\frac{\partial \mathbf{r}}{\partial u}\cdot\frac{\partial \mathbf{r}}{\partial u}\right]_0. \qquad 19$$

The subscript, 0, is to indicate the unstrained condition so that L_u is the length of member per unit u in the unstrained condition. Engineers may think that equation 18 is rather an odd definition of strain, but if strain is small in the directions of the members, then this definition gives the same result as the more usual definition. Equation 18 has a number of advantages, including the fact that it does not require a square root.

Corresponding to equations 13, 15, 16, 17, 18 and 19 there are equations giving the curvatures, twist and axial strain of the members running in the direction of increasing v.

The strain in the ties,

$$\varepsilon_{tie} = \frac{1}{2}\left[\frac{\left[\frac{\partial \mathbf{r}}{\partial u} \pm \frac{\partial \mathbf{r}}{\partial v}\right]\cdot\left[\frac{\partial \mathbf{r}}{\partial u} \pm \frac{\partial \mathbf{r}}{\partial v}\right]}{\left[\frac{\partial \mathbf{r}}{\partial u} \pm \frac{\partial \mathbf{r}}{\partial v}\right]_0\cdot\left[\frac{\partial \mathbf{r}}{\partial u} \pm \frac{\partial \mathbf{r}}{\partial v}\right]_0} - 1\right]. \qquad 20$$

The + or - is taken according to which diagonal direction a tie follows.

Equation 6 gives $\mathbf{r}$ whose components are the x, y and z coordinates of a typical point on an element. If the coordinates of the control points change slighly, then $\mathbf{P}$ (equation 8) will change by $\delta\mathbf{P}$ and $\mathbf{r}$ will change by

$$\delta\mathbf{r} = \begin{bmatrix}1 & u & u^2 & u^3\end{bmatrix}\mathbf{D}\,\delta\mathbf{P}\mathbf{D}^T\begin{bmatrix}1\\ v\\ v^2\\ v^3\end{bmatrix}. \qquad 21$$

Differentiation of equations 15, 16, 17 and 18 (and the corresponding equations for the v direction) and of equation 20 enable the increments of curvature, twist and axial strain to be calculated as quantities dependent upon δP.

Finally constants describing the bending, twisting and axial stiffness of grid members and the axial stiffness of ties are required. In calculating the axial force in ties a condition is necessary stating that they cannot take compression.

Simple experiments with grid shells show that the axial stiffness of grid members (but not ties) is so high that the actual value of stiffness is immaterial. For this reason it was decided that it is more elegant to replace the usual

$$\text{tension} = \text{Young's modulus} \times \text{cross-sectional area} \times \text{strain} \qquad 22$$

by the equation

$$\text{axial strain in grid members} = 0 \qquad 23$$

which leaves the value of tension indeterminate, at least temporarily.

Finite element formulation

Let us write the x, y and z coordinates of the control points defining all the elements describing a grid shell in a column matrix, $\mathbf{X}$. In order to do this we will require some overall numbering system for the control points in the same way that nodes are numbered in the finite element method.

Corresponding to $\mathbf{X}$ there will be a column matrix, $\mathbf{F}$, containing the components of the out of balance 'forces' applied to each control point. A typical component of $\mathbf{F}$, say F_i, is found by using a compatibility set in which only the i^{th} component of $\mathbf{X}$, X_i (which could be the x, y or z coordinate of a control point), changes by a small amount.

F_i multiplied by the change in X_i is equal to the left hand side of the virtual work equation, equation 12. Thus the out of balance 'forces' are the applied 'loads' minus the 'resistance' from the structure. F_i will only equal zero for all values of i, that is

$$\mathbf{F} = \mathbf{0}, \qquad 24$$

when we have achieved equilibrium.

The tensions in 'selected' grid shell members need to be stored in a column matrix which we shall call $\mathbf{T}$. We will return to the question of what exactly we mean by 'selected'. The strain in the same members will be stored in the column matrix $\mathbf{e}$. The strains are defined in such a way that

$$\mathbf{T}^T \delta \mathbf{e} = \text{virtual work absorbed by tensions in grid shell members} \qquad 25$$

Note that we can use equation 25 as part of the build up of the finite element formulation despite the fact that we shall subsequently impose the condition

$$\mathbf{e} = \mathbf{0} \qquad 26$$

which corresponds to our assumption that grid shell members (but not ties) have infinite axial stiffness. This is because our compatibility sets are 'hypothetical'.

We also need to consider boundary constraints. If $\mathbf{D}$ is a column matrix of boundary displacements and rotations and $\mathbf{R}$ is a column matrix of forces and moments exerted by the shell on its boundary supports, then $\mathbf{R}$ and $\mathbf{D}$ have to be defined such that

$$\mathbf{R}^T \delta \mathbf{D} = \text{virtual work done by boundary thrusts and moments.} \qquad 27$$

Again we can use equation 27 despite the fact that we shall subsequently impose the condition

$$\mathbf{D} = \mathbf{0}. \qquad 28$$

In order to achieve our desired deformed state of equilibrium with zero axial strain in grid members and satisfaction of boundary constraints, we have to satisfy equations 24, 26 and 28, all at the same time. To find such a state we require equations which tell us how $\mathbf{F}$, $\mathbf{e}$ and $\mathbf{D}$ change due to changes in $\mathbf{X}$, $\mathbf{T}$ and $\mathbf{R}$. The change in $\mathbf{F}$ is caused by changes in each of $\mathbf{X}$, $\mathbf{T}$ and $\mathbf{R}$ so that, in matrix notation

$$\delta \mathbf{F} = -\mathbf{K} \delta \mathbf{X} + \delta \mathbf{F}_{\text{Tensions}} + \delta \mathbf{F}_{\text{Support reactions}}$$

where

$$\delta \mathbf{F}_{\text{Tensions}} = -\mathbf{H} \delta \mathbf{T} \qquad 29$$

and

$$\delta \mathbf{F}_{\text{Support reactions}} = -\mathbf{Y} \delta \mathbf{R}.$$

$\mathbf{K}$ is a square matrix and the size and shape of $\mathbf{H}$ and $\mathbf{Y}$ correspond to the number of elements in the column matrices, $\mathbf{F}$, $\mathbf{T}$ and $\mathbf{R}$.

The changes in $\mathbf{e}$ and $\mathbf{D}$ depend only upon $\delta\mathbf{X}$ so that

$$\delta\mathbf{e} = \mathbf{J}\,\delta\mathbf{X} \qquad 30$$

and

$$\delta\mathbf{D} = \mathbf{B}\,\delta\mathbf{X}. \qquad 31$$

Space does not permit a discussion of the details of how $\mathbf{K}$, $\mathbf{H}$, $\mathbf{Y}$, $\mathbf{J}$ and $\mathbf{B}$ are determined. However their derivation is a purely mechanical one involving differentiation and numerical integration which is required to evaluate the left hand side of the virtual work equation, equation 12.

However, things are not quite as complicated as they might seem since

$$\mathbf{J} = \mathbf{H}^T \qquad 32$$

and

$$\mathbf{B} = \mathbf{Y}^T. \qquad 33$$

These equations follow from application of the virtual work equation to give:

$$\left[\delta\mathbf{F}_{\text{Tensions}}\right]^T \delta\mathbf{X} = -\left[\delta\mathbf{T}\right]^T \delta\mathbf{e} \qquad 34$$

and

$$\left[\delta\mathbf{F}_{\text{Support reactions}}\right]^T \delta\mathbf{X} = -\left[\delta\mathbf{R}\right]^T \delta\mathbf{D}. \qquad 35$$

Hence

$$-\left[\mathbf{H}\,\delta\mathbf{T}\right]^T \delta\mathbf{X} = -\left[\delta\mathbf{T}\right]^T \mathbf{J}\,\delta\mathbf{X} \qquad 36$$

and

$$-\left[\mathbf{Y}\,\delta\mathbf{R}\right]^T \delta\mathbf{X} = -\left[\delta\mathbf{R}\right]^T \mathbf{B}\,\delta\mathbf{X}. \qquad 37$$

which give equations 32 and 33.

The final equation is obtained by combining equations 29, 30, 31, 32 and 33 to give

$$\begin{bmatrix} -\delta\mathbf{F} \\ \delta\mathbf{e} \\ \delta\mathbf{D} \end{bmatrix} = \mathbf{Z} \begin{bmatrix} \delta\mathbf{X} \\ \delta\mathbf{T} \\ \delta\mathbf{R} \end{bmatrix} \qquad 38$$

in which the square matrix, $\mathbf{Z}$ is given by

$$\mathbf{Z} = \begin{bmatrix} \mathbf{K} & \mathbf{H} & \mathbf{Y} \\ \mathbf{H}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{Y}^T & \mathbf{0} & \mathbf{0} \end{bmatrix}. \qquad 39$$

The complete procedure to find the deformed shape in equilibrium with given loads is

1. Assume a starting configuration defined by initial values of $\mathbf{X}$, $\mathbf{T}$ and $\mathbf{R}$

2. Calculate the out of balance forces, $\mathbf{F}$, the grid member elongations, $\mathbf{e}$, and the displacements and rotations at boundary constraints, $\mathbf{D}$, which should all be zero, but will not be so at the start of the analysis.

3. Set

$$\begin{bmatrix} -\delta\mathbf{F} \\ \delta\mathbf{e} \\ \delta\mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ -\mathbf{e} \\ -\mathbf{D} \end{bmatrix}. \qquad 40$$

4. Solve equation 38 for $\delta\mathbf{X}$, $\delta\mathbf{T}$ and $\delta\mathbf{R}$

5. Set

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{T} \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{X} \\ \mathbf{T} \\ \mathbf{R} \end{bmatrix} + \begin{bmatrix} \delta\mathbf{X} \\ \delta\mathbf{T} \\ \delta\mathbf{R} \end{bmatrix}. \qquad 41$$

6. Go back to step 2 and continue until $\mathbf{F}$, $\mathbf{e}$ and $\mathbf{D}$ are acceptably small.

Clearly the whole proceedure requires that the square matrix $\mathbf{Z}$ in equation 38 is reasonably well conditioned. We will discuss this question in the next sec-

tion.

Geometrical Constraints

Equations 26 and 28 are geometrical constraints which reduce the effective number of degrees of freedom. Clearly we cannot have so many geometric constraints that the whole structure locks solid. A single bicubic B-spline patch has 16 contol points and therefore $16 \times 3 = 48$ degrees of freedom. However in a surface that is modelled by a reasonably large number of patches, the sharing of control points by adjacent patches means that away from the edge of the structure the number of control points is equal to the number of patches. Thus effectively there are only 3 degrees of freedom per patch. This means that there can be no more than two geometric constraints per patch forcing the grid member strains to be zero.

The obvious location for these constraints is along the four sides of each patch. This is only two constraints per patch since each side is shared by two patches.

Boundary constraints are more difficult to discuss in general terms, but again one must not impose too many geometric constraints.

Example

Figure 3 shows a computer plot of an unloaded grid shell modelled by four bicubic B-spline patches. There are no ties on the structure. Figure 4 shows the deformed shape of the numerical model caused by application of a uniformly distributed vertical load to the entire shell. Figure 5 shows the deformed shape caused by application of a uniformly distributed load to one quarter of the shell.

Figure 6 shows a photograph of a physical model of the same structure subject to a number of point loads intended to model the uniformly distributed load over a quarter of the shell used to produce figure 5. The model is made from a woven mild steel wire mesh.

Quantitative comparison of results from the computer and physical models showed very good agreement. The computer model overestimated the collapse load by approximately 10%. This could be due to the fact that yielding of the steel wires began prior to final collapse of the physical model.

The above theoretical description introduced 'rigid' boundary constraints via equation 28. The computer analysis which produced figures 3, 4 and 5 actually used stiff springs instead of rigid restraints, basically because the rigid boundary formulation had not been derived at that time.

The computer model did include the condition of zero axial strain in the grid members (equation 26). However since there are only four elements and $5 \times 5 = 25$ control points and therefore 75 degrees of freedom, it was possible to make the zero strain condition on 8 members per element, rather than the 2 (or 4 if shared) discussed above.

Conclusions and suggestions for further work

The limited comparison of computer results and physical model tests suggests that the bicubic B-spline finite element can be used to produce accurate predictions of the buckling loads on a grid shell.

However, as mentioned in the introduction, the authors would not recommend the use of a computer analysis as the sole means of justication of the structural safety of grid shells. This is partly because of the insight that a physical model gives and partly because one can only 'trust' a computer program after it has been thoroughly tested in a wide range of application.

Grid shells made of timber are subject to creep and possibly creep buckling - a slowly accelerating collapse involving buckling and creep. This is a horribly complicated matter which could, perhaps, be included in a bicubic B-spline based computer model.

Finally in double layer grid shells like those at Mannheim, there is the question of relative slip between the two layers of grid running in parallel directions.

References

1. Otto, F.: *IL 10*, *Gitterschalen*, *Grid Shells*, Institut für leichte Flächentragwerke, Universität Stuttgart, 1974.

2. Happold, E. and Liddell, W. I.: 'Timber lattice roof for the Mannheim Bundesgartenschau', *The Structural Engineer*, **53**, No. 3, March 1975, pp. 99 - 135.

3. Otto, F.: *IL 13, Multihalle, Mannheim*, Institut für leichte Flächentragwerke, Universitat Stuttgart, 1978.

4. Faux, I. D. and Pratt, M. J.: *Computational geometry for design and manufacture*, Ellis Horwood, 1979.

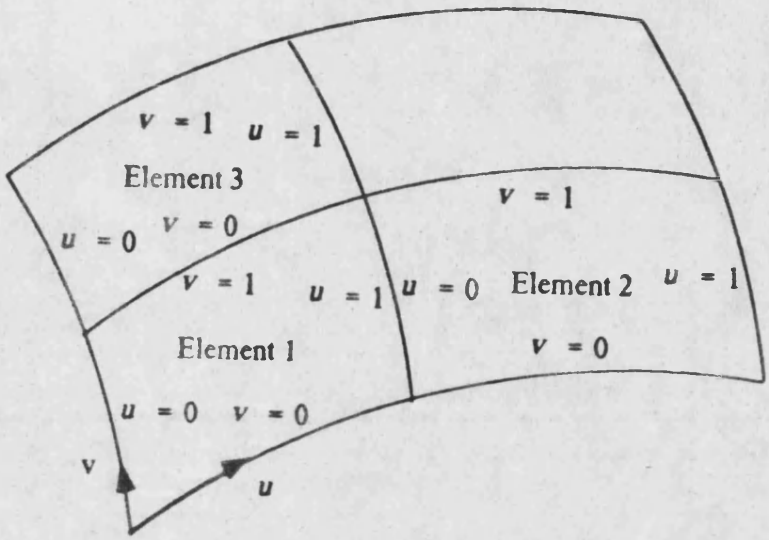

v = 1
u = 1
Element 3
u = 0
v = 0
v = 1
u = 1
u = 0
Element 2
u = 1
v = 1
Element 1
v = 0
u = 0
v = 0
v
u

Figure 1

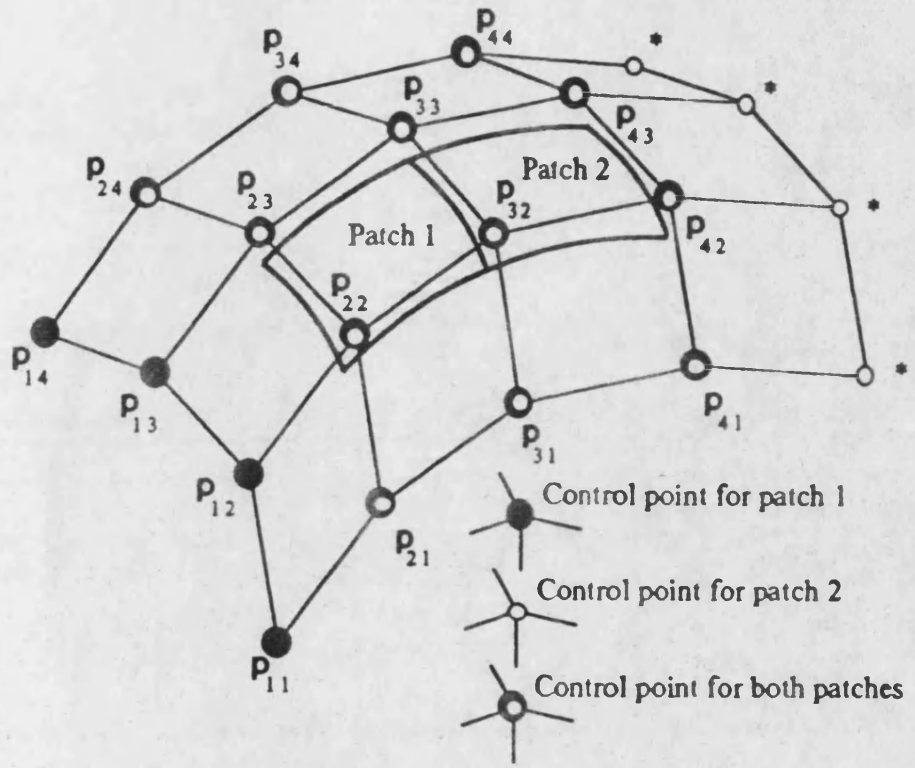

P34
P44
P33
P43
P24
P23
Patch 2
P32
Patch 1
P42
P22
P14
P13
P41
P31
P12
P21
P11
Control point for patch 1
Control point for patch 2
Control point for both patches

Figure 2

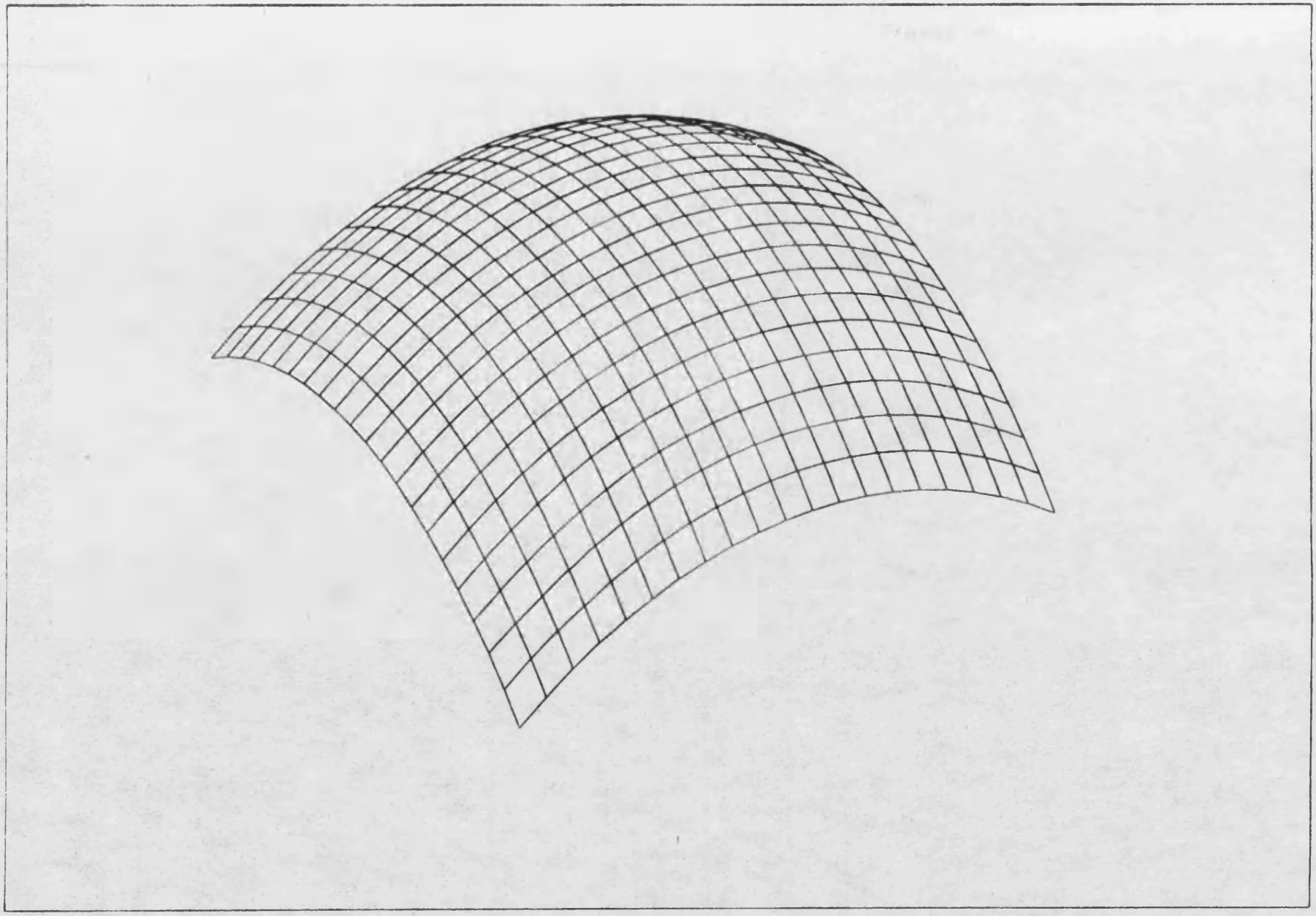
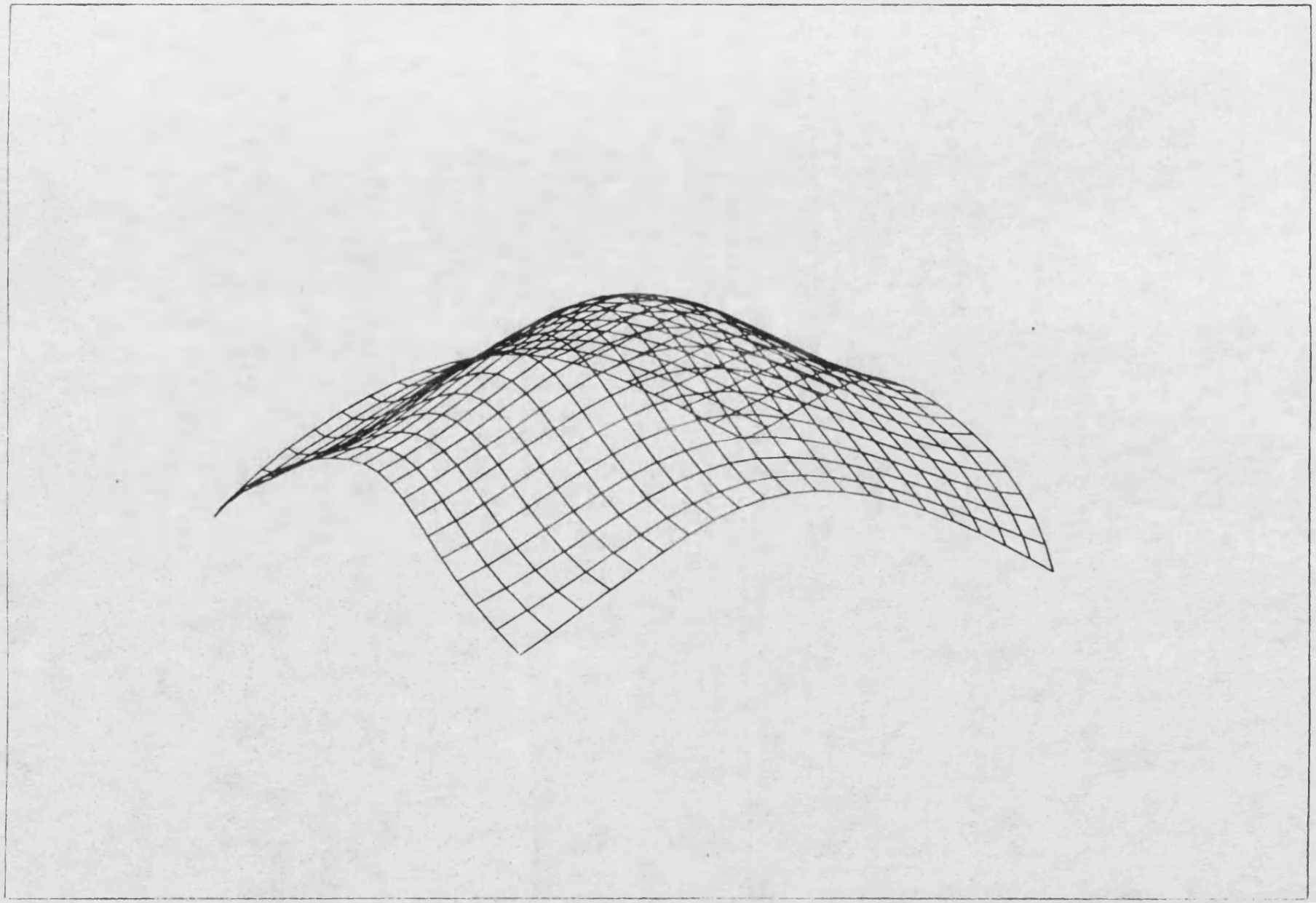


FIGURE 3

FIGURE 4



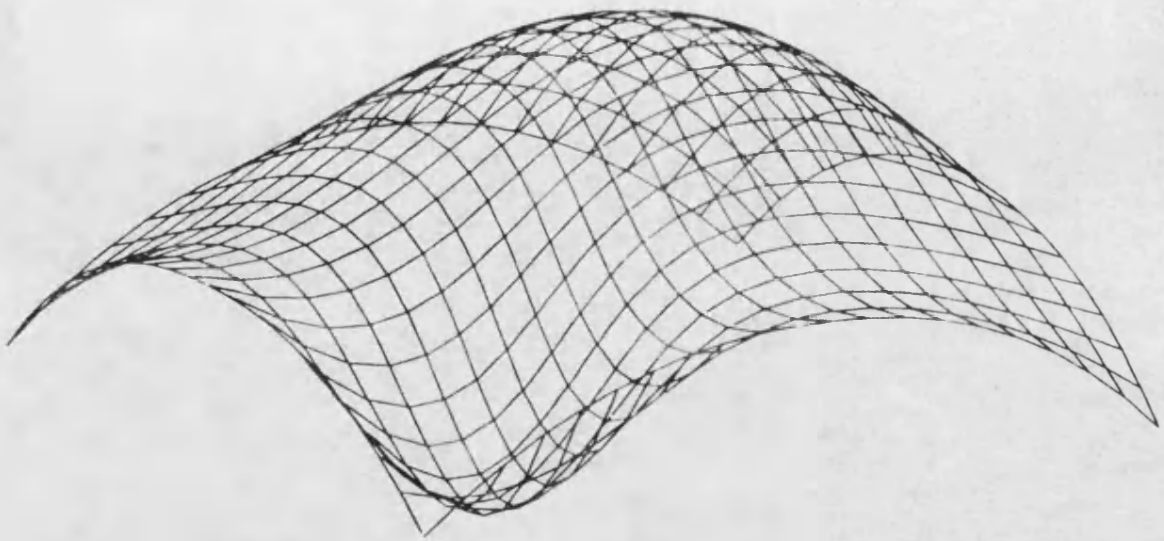


FIGURE 5

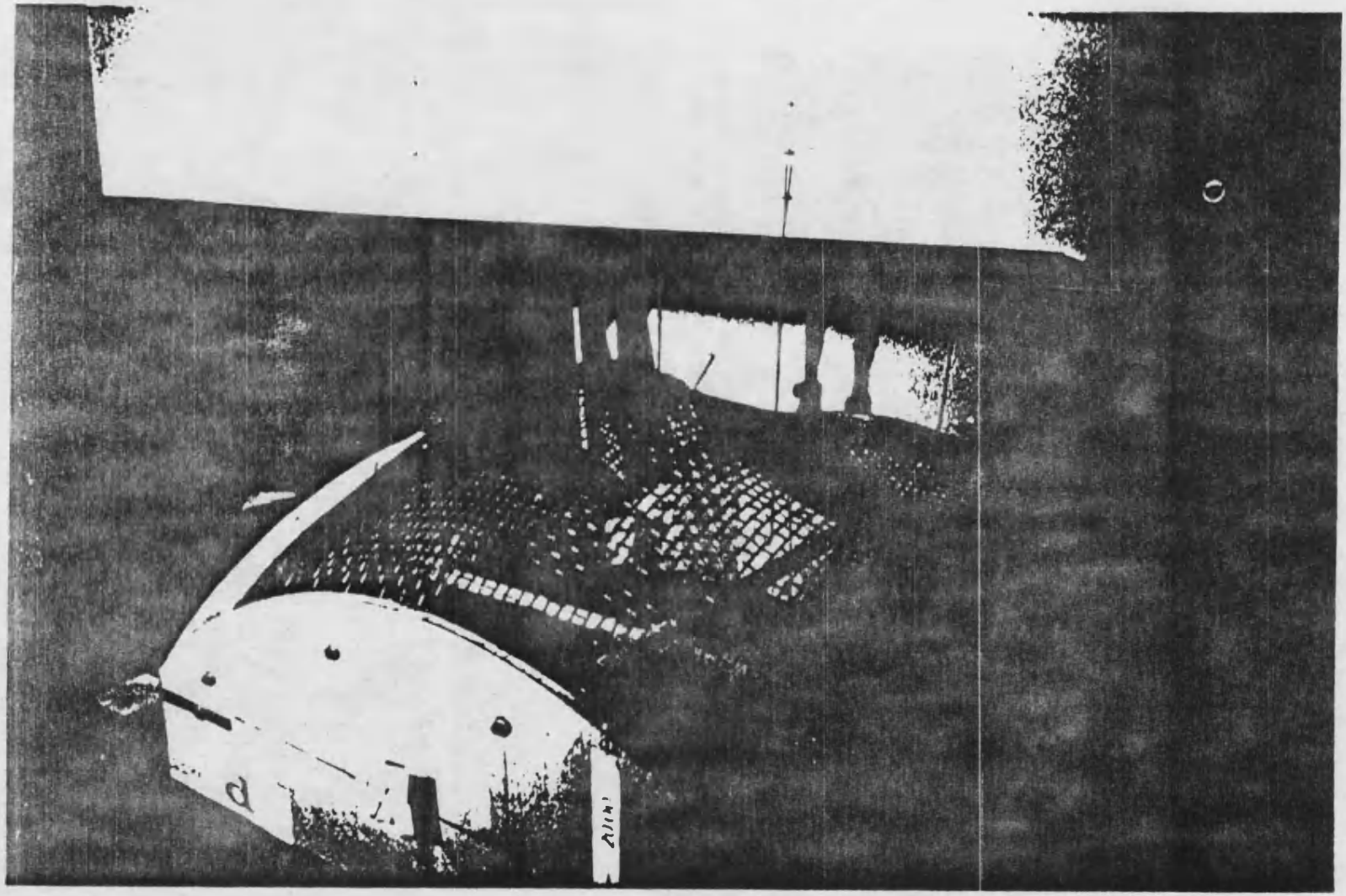


FIGURE 6.

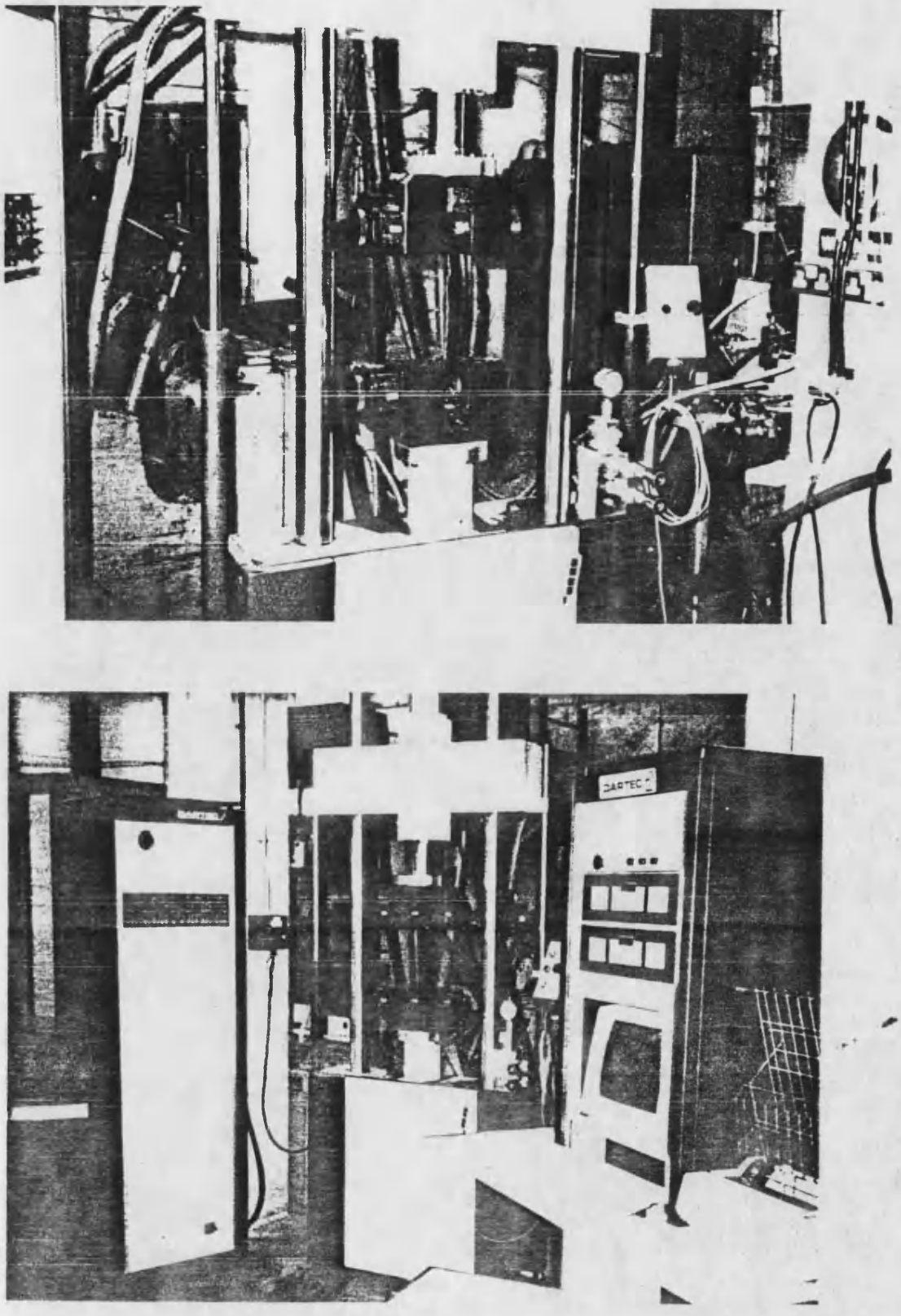


Fig.(7-8) The tension test using the 10 t DARTEC machine

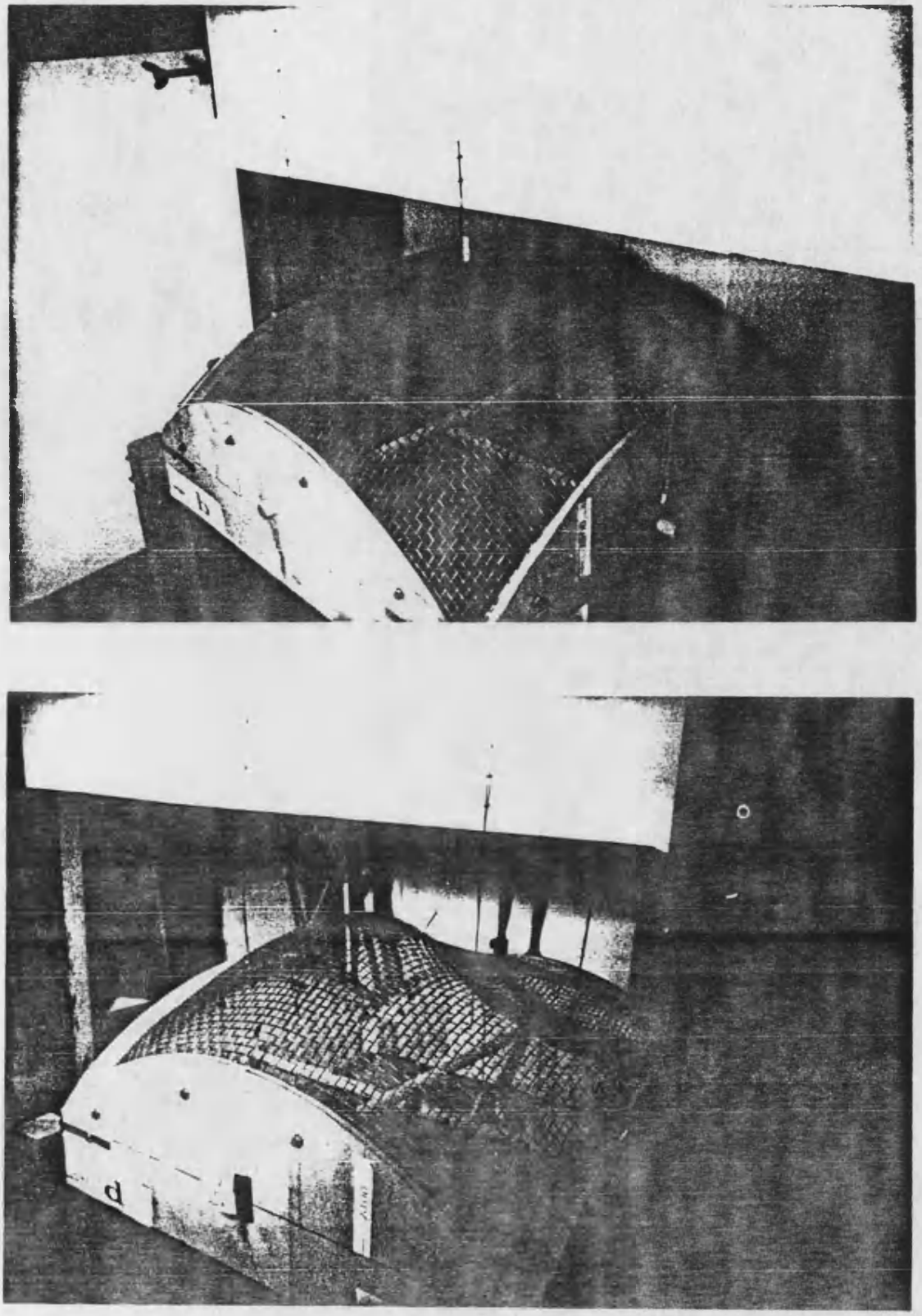
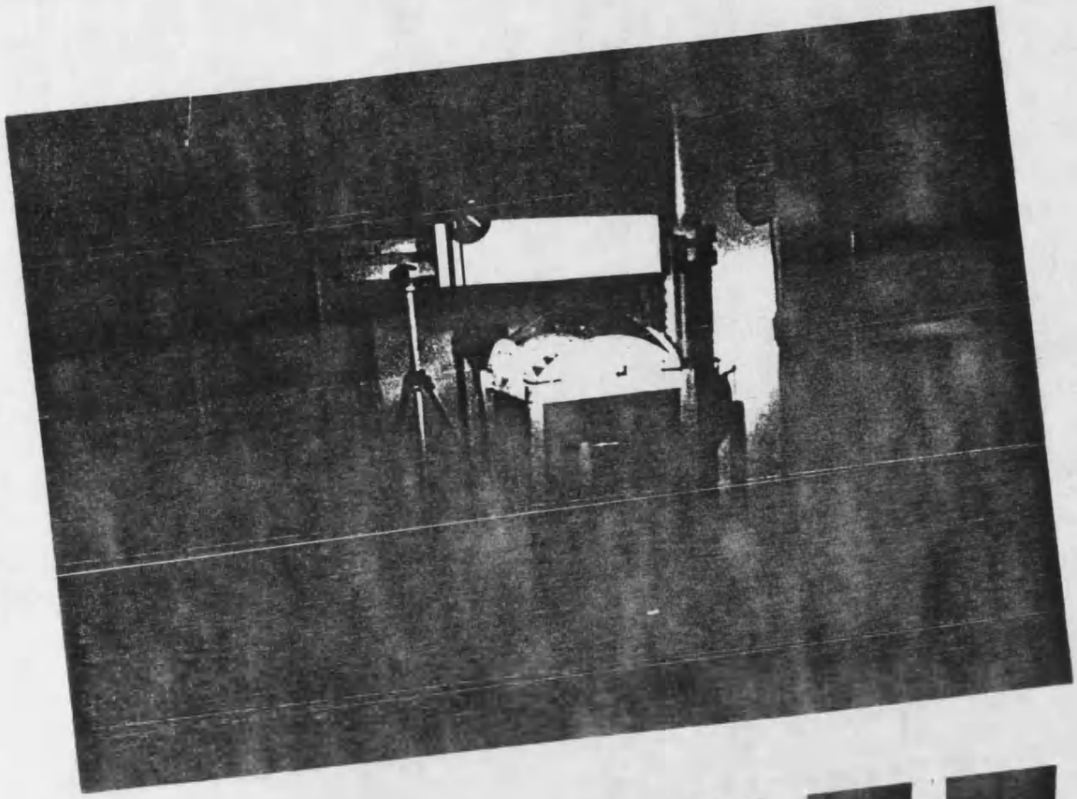


Fig.(7-14) The unsymmetrical loading



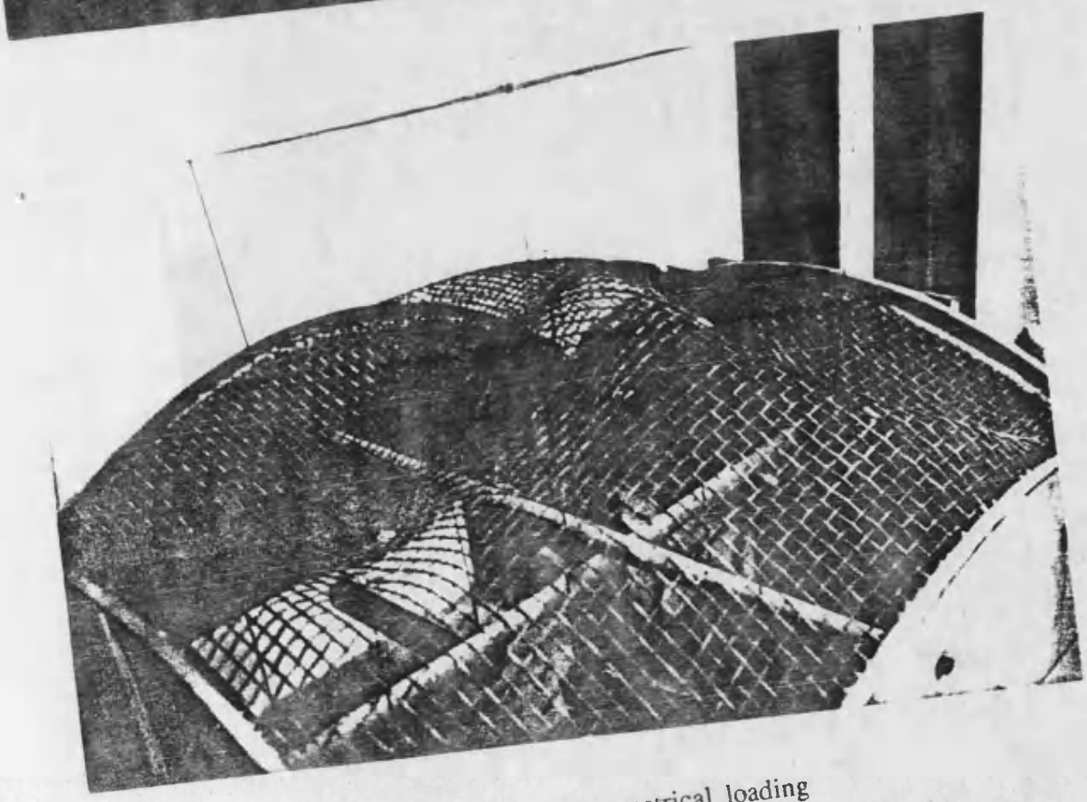


Fig.(7-15) Symmetrical loading

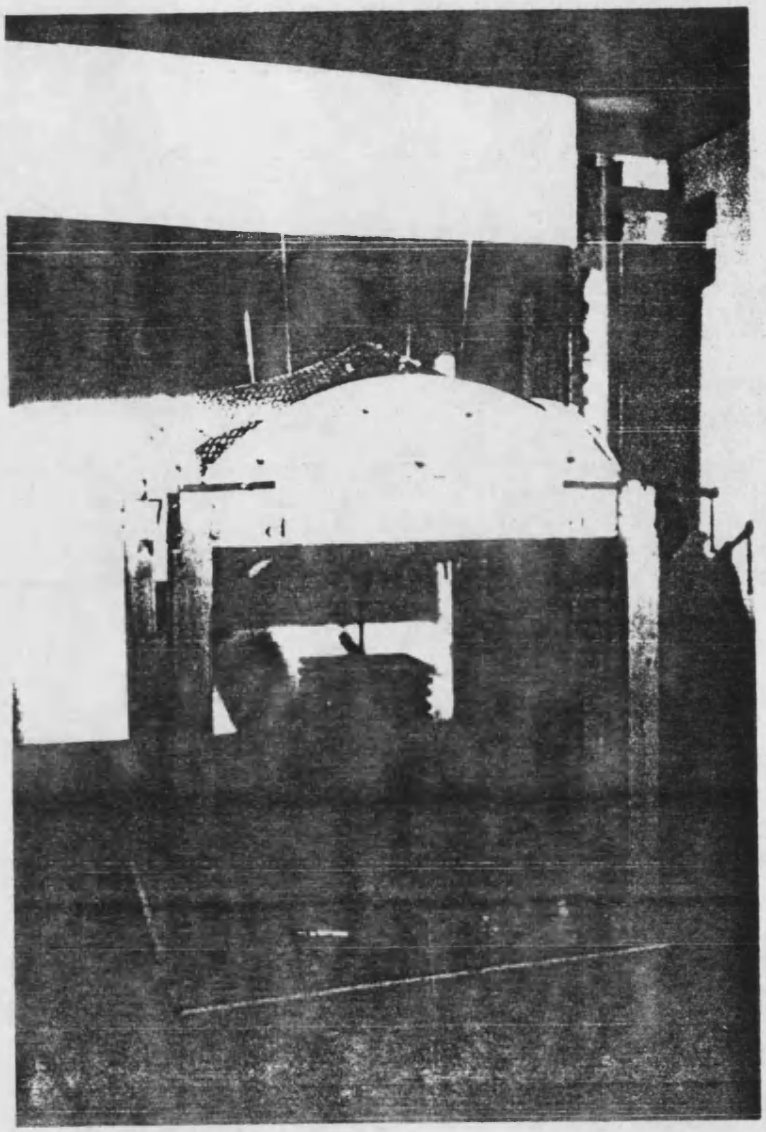


Fig.(7-16) The crossed resistance lines against the deflection